Robust Sparse Recovery via Matching Pursuit Algorithms and applications to simultaneous-source seismic data processing

by

Ji Li

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science
in
Geophysics

Department of Physics
University of Alberta

Abstract

Compressive sensing and sparse reconstruction techniques are adopted to solve many seismic data processing problems, including the design of high-resolution transforms for coherent noise removal, signal separation, and seismic wavefield reconstruction. Traditional sparse reconstruction algorithms optimally work with noise-free data or data contaminated with random noise of Gaussian distribution. Their performance degrades in the presence of erratic noise. In this thesis, erratic noise refers to noise characterized by large and isolated amplitudes. The thesis proposes robust sparse reconstruction algorithms that are resistant to erratic noise. These algorithms are adopted to solve the simultaneous-source separation problem via robust sparse Radon transforms.

I first review different robust sparse reconstruction algorithms based on classical M-estimators. Then, I propose a robust Matching Pursuit (MP) algorithm to retrieve sparse Radon domain coefficients. The algorithm is robust to outliers and, hence, applicable to process simultaneous-source seismic data. The proposed robust MP algorithm is slow when applied to synthetic data and field data. Thus, I introduce the robust Stagewise Weak Conjugate Gradient Pursuit (SWCGP), which reduces the total costs of robust MP by selecting multi-coefficients at each iteration and performs a simple conjugate gradient optimization step in each iteration. The results show that the robust SWCGP can use much fewer iterations to achieve similar results as robust MP.

Preface

Chapters 3, and 4 in this thesis are original work. Chapter 3 contains the material published in J Li and M D Sacchi, 2021, An lp-space Matching Pursuit algorithm and its application to robust seismic data denoising via time-domain Radon transforms: Geophysics, in Press.

In these publications, I was responsible for designing and programming the algorithms, preparing data examples and writing the manuscripts. Dr. Sacchi was the supervisory author and was involved in concept formulation and manuscript editing.

Acknowledgements

First, I want to thank my supervisor, Dr. Mauricio Sacchi, for his patience and support during my research and completion of my MSc degree. I also want to thank all of my colleagues in the Signal Analysis and Imaging Group (SAIG). I feel honored to be a part of this group. I would also want to thank all the sponsors for providing financial support for our group.

Contents

Appendices

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Compressive sensing

Compressive Sensing is a topic that has received considerable attention in geophysics in the last decade. It was first introduced in 2006 by Donoho (2006); Candes et al. (2006). And since then, thousands of papers have been published in different areas such as seismic exploration (Hennenfent and Herrmann, 2008; Herrmann, 2010), medical imaging (Lustig et al., 2007), remote sensing (Ahmad and Amin, 2013), radar imaging (Baraniuk and Steeghs, 2007), and astronomical imaging (Bobin et al., 2008). In geophysics, applications of Compressive Sensing can be grouped into two categories: acquisition design and data reconstruction.

A fundamental principle in signal processing is given by Nyquist-Shannon sampling theory (Nyquist, 1928; Shannon, 1949). According to the Nyquist-Shannon sampling theorem, a signal can be exactly recovered from a set of uniformly sampled data taken at the so-called Nyquist rate, which is at least twice the highest frequency the data itself. It means that if we sample a signal in which the highest frequency we want to reconstruct correctly is f, then we must sample the signal at a minimum frequency of $2f$. Sampling the signal at a frequency lower than the Nyquist frequency will cause aliasing. It is sometimes unrealistic for geophysical data acquisition to sample the data with density prescribed by Nyquist-Shannon sampling. This is particularly true for spatial sampling, where one must deploy receivers and sources on the surface of Earth to acquire seismic data. In general, such sampling can lead to a costly acquisition of the seismic wavefield. Compressive Sensing states that a finite-dimensional signal having a sparse or compressible representation can be recovered from a small set of linear, non-adaptive measurements (Baraniuk, 2007; Candes and Wakin, 2008). Compressive Sensing for geophysical applications is based on two key elements. First, the acquisition or sampling is random; second, the signal is sparse under a particular transform like the Fourier transform, wavelet transform, or is constrained by a low-rank constraint when the desired signal can be embedded in a matrix.

Figure 1.1 shows a simple example showing the principles of Compressive Sensing. Figure 1.1(a) is a complex signal of wavenumber $k = 0.3$ radians/unit. The green signal is sampled every $\Delta x = 1$ units and yields 11 regularly sampled observations given by red dots. Figure 1.1(b) is the frequency spectrum calculated using sparse inversion. We observe one amplitude in the frequency spectrum corresponding to the wavenumber $k = 0.3$ radians/unit. Figure 1.1 (c) is the reconstructed signal estimated from the inverted Fourier coefficients. Figure 1.1 (d) shows the signal sampled at $\Delta x = 2$ units which corresponds to extracting 6 equally spaced observations. Because the sampling rate is below the Nyquist rate, aliasing happened. We find that there are two peaks in the frequency spectrum portrayed by Figure 1.1e. Therefore, the reconstructed result in Figure 1.1 (f) corresponds to the superposition of two harmonics signals. Clearly, this is a consequence of sampling the green signal in Figure 1.1d at a rate that corresponds to less than two points per cycle. The recovered data (blue curve) shown in Figure 1.1f reproduces the observations (red dots) but the fit differs significantly from the original green curve in Figure 1.1d. Figure 1.1 (g) corresponds to the random sampling case. The signal in green was sampled by 6 observations as in the aliased case, but now the 6 regularly sampled positions where perturbed to generate random sampling. The average sampling rate is below the Nyquist rate, but the sparse inversion algorithm has now retrieved one spectral peak at the correct wavenumber $k = 0.3$ radians/unit (Figure 1.1 (h)). The estimated Fourier coefficients are then used to reconstruct the signal in blue portrayed by Figure 1.1(i). The idea of compressive reconstruction is to recover the Fourier coefficients that synthesize the observations via regularized sparse inversion. The regularization term of the inversion is a sparsity promoting term. When this concept is combined with random sampling, one can sample at a rate that is more economical that the one provided by classical sampling theory based on Nyquist-Shannon theory.

This thesis will focus on the sparse recovery theory and on algorithms for sparse signal reconstruction. We will pay specific attention to robust reconstruction algorithms for sparse signals when erratic noise corrupts data.

1.2 Recovery Algorithms

Generally speaking, according to the theory of basis pursuit (BP) (Chen et al., 1998), signals can often be well-approximated as a linear combination of just a few elements from a known basis or dictionary. For a signal $\mathbf{x} \in \mathbf{R}^n$ we call the signal $\mathbf{x}$ is k-sparse when it has at most k non-zero elements, i.e.,$\|\mathbf{x}\|_0 \leq k$, and $k \ll n$. Therefore, given the observed data $\mathbf{y}$, if we attempt to reconstruct the signal $\mathbf{x}$, it

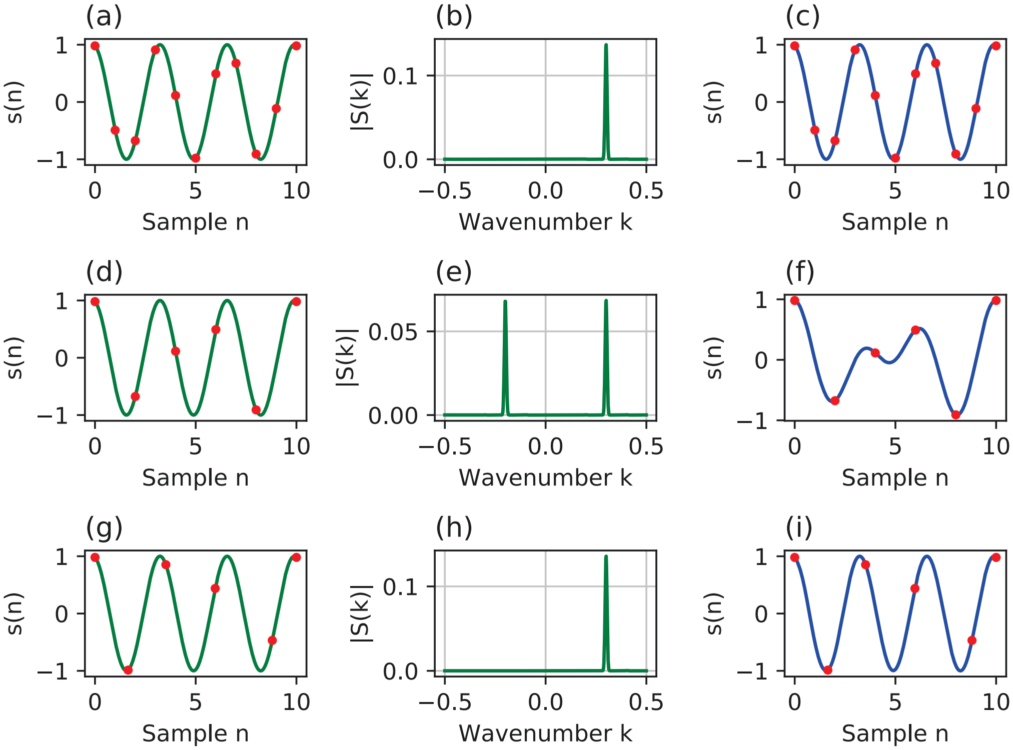

Figure 1.1: (a) Signal sampled at Nyquist rate $\Delta x = 1$ unit. (b) Frequency spectrum of (a) estimated via sparse inversion. (c) Reconstructed signal (blue) synthesized from the inverted Fourier coefficients. (d) Signal sampled at half of Nyquist rate $\Delta x = 2$ units. (e) Frequency spectrum of (d) estimated via sparse inversion. (f) Reconstructed signal (blue) via Fourier synthesis. (g) Signal sampled randomly. (h) Frequency spectrum of (g) estimated via sparse inversion. (i) Reconstructed signal via Fourier synthesis. This example corresponds to a complex signal of single wavenumber $k = 0.3$ radians/unit. For illustrative purposes the real part of the signals were displayed.

is straightforward for us to adopt the following minimization problem

$$\underset{\mathbf{x}}{min} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 = 0\,. \tag{1.1}$$

In real data situations, the observed data are contaminated by noise. In this situation, the optimization problem becomes

$$\underset{\mathbf{x}}{min} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{A}\mathbf{x}-\mathbf{y}\|_2^2 \leq \epsilon\,, \tag{1.2}$$

where $\epsilon > 0$ is used to bound the ℓ_2-norm of the residual error and is pre-determined by the noise level. In general, solving the nonconvex problems given by equation 1.2 is known to be NP-hard and intractable. The problem can be solved when the sensing matrix $\mathbf{A}$ obeys the restricted isometry property (RIP) (Candès, 2008).

Definition 1.2.1. A matrix $\mathbf{A}$ satisfies the restricted isometry property (RIP) of order k if there exist a constant $\delta_k \in$ (0,1), defined as the smallest positive quantity such that

$$(1-\delta_k)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1+\delta_k)\|\mathbf{x}\|_2^2 \tag{1.3}$$

holds for all $\mathbf{x} \in \sum_k$ where $\sum_k = \{x \in \mathbb{R}^n | \quad \|x\|_0 \leq k\}\,.$

If the condition given by the RIP property is satisfied, then various algorithms can recover the sparse signal from noisy measurements. Generally, there are two different kinds of approaches to solving this optimization problem. The first approach is based on convex relaxation, which uses the ℓ_1-norm to replace the nonconvex ℓ_0-norm, therefore, making the problem convex. The second approach for solving the sparse approximation problem is greedy pursuit.

1.2.1 Convex relaxation methods

The first approach is based on convex relaxation. The convex relaxation method uses the ℓ_1-norm to replace the ℓ_0-norm in the equation 1.2. Then the minimization problem becomes

$$\underset{\mathbf{x}}{min} \, \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \leq \epsilon \, , \tag{1.4}$$

which is also known as Basis Pursuit denoising (BPDN) (Chen et al., 1998). This constrained optimization problem can be converted into an alternative unconstrained form, which is known as LASSO (least absolute shrinkage and selection operator) (Tibshirani, 1996)

$$\underset{\mathbf{x}}{min} \, \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1, \tag{1.5}$$

where $\lambda > 0$ is a regularization parameter that controls a tradeoff between the residual error term and the regularization term. Under certain conditions, the ℓ_1-norm minimization solution problem coincides with that of the ℓ_0-norm minimization problem. An important advantage of adopting an ℓ_1-norm regularization is that problems become tractable by classical optimization tools for convex problems. The ℓ_1-regularized least squares (LASSO) is sometimes preferred over BPDN because of the availability of efficient methods to solve the problem given by equation 1.5. However, in generally, the value of λ is unknown. Moreover, the solution of the problem given by equation 1.5 is highly sensitive to the value of λ. Several authors discuss the selection of the tradeoff scalar λ (Eldar, 2008; Galatsanos and Katsaggelos, 1992; Golub et al., 1979) and, in general, trade-off parameter selection can be a challenge for large-scale applications where one may not be able to explore solutions for different λ values.

Many algorithms are available to solve the ℓ_1 regularization problem. The projected gradient method, for instance, is a method that seeks sparse a representation of $\mathbf{x}$ along a given gradient direction. There exist two slightly different versions of it, one is the gradient projection sparse representation (GPSR) (Figueiredo et al., 2007), and the other is the truncated Newton interior-point method (TNIPM) (Kim et al., 2007). Homotopy methods (Osborne et al., 2000; Malioutov et al., 2005) are another suitable approach for large-scale sparse solvers. For instance, if a k-sparse signal is sufficiently sparse, the homotopy methods can find it in k iterations with high probability. Another algorithm is called the iterative shrinkage-thresholding algorithm (ISTA) (Combettes and Wajs, 2005; Daubechies et al., 2004). ISTA updates the solution via "soft-thresholding" function (Donoho, 1995).

Beck and Teboulle (2009) proposed the fast iterative shrinkage thresholding algorithm (FISTA), which is an improvement of ISTA. FISTA also improves the sequence of iteration points; instead of employing the previous point, it utilizes specific linear combinations of the previous two points.

The alternating direction method of multipliers(ADMM) is another powerful algorithm suitable for large-scale machine learning and signal processing problems, developed long ago and reviewed recently by Boyd et al. (2011). ADMM has also been adopted to solve linear problems with ℓ_1 regularization.

1.2.2 Greedy methods

The second group of approaches can be summarized as a group of greedy methods. Generally, greedy techniques can be divided into two categories. The first set of strategies is called "greedy pursuit," which can be defined as a set of techniques that iteratively build up an estimate of the signal $\mathbf{x}$. The second set of approaches called

”Thresholding” algorithms. These methods are often straightforward to implement and can be relatively fast.

Greedy pursuit

Mallat and Zhang (1993) introduced the idea of greedy pursuit in signal processing. Greedy pursuits methods share the two basic steps: element selection and coefficient update. These methods are usually initialized with a zero estimate, $\hat{\mathbf{x}}^{[0]} = \mathbf{0}$. With this initialization, the initial residual error is $\mathbf{r}^{[0]} = \mathbf{y} - \mathbf{A}\hat{\mathbf{x}}^{[0]} = \mathbf{y}$ and the support set (i.e. the indices of the nonzero elements) of the first estimate $\hat{\mathbf{x}}^{[0]}$ is $\mathbf{T} = \emptyset$. Each iteration then updates these quantities by adding additional elements (columns from $\mathbf{A}$) to the support set $\mathbf{T}$ and updating the signal estimate $\hat{\mathbf{x}}$, thereby decreasing the residual observation error $\mathbf{r}$.

Matching Pursuit (MP) (Mallat and Zhang, 1993) is one of the simplest pursuit algorithms that one can code. The approximation is incremental, selecting one column from $\mathbf{A}$ at a time, and, at each iteration, only the coefficients associated with the selected column of the dictionary $\mathbf{A}$ is updated. MP will stop in a finite number of iterations if the norm of $\mathbf{r}^{[i]}$ is used to define a stopping criterion for the algorithm. The dominant computational cost of MP arises from the repeated evaluation of matrix-times-vector multiplications involving $\mathbf{A}^T$. Therefore MP is generally proposed for applications where operations with $\mathbf{A}$ and $\mathbf{A}^T$ admit a fast implementation in implicit form such as in case of the Fast Fourier Transform (FFT).

A more sophisticated strategy is implemented by Orthogonal Matching Pursuit (OMP) (Pati et al., 1993; Tropp and Gilbert, 2007; Tropp, 2004). In OMP the approximation for $\mathbf{x}$ is updated in each iteration by projecting $\mathbf{y}$ orthogonally onto the columns of $\mathbf{A}$ associated with the current support set $\mathbf{T}^{[i]}$. Note that in contrast to MP, the minimization is performed with respect to all of the currently selected coefficients. Unlike MP, OMP never re-selects an element, and the residual at any iteration is always orthogonal to all currently selected elements. While OMP is more computationally complex than MP due to the orthogonalization step, it generally enjoys superior performance compared to the MP.

One problem with MP and OMP type strategies is the need to perform at least as many iterations as there are atoms (elements from the dictionary $\mathbf{A}$) to be selected. Unfortunately, this precludes MP and OMP to be adopted for large scale problems. To speed up pursuit these algorithms, it is thus necessary to select multiple elements at a time. This idea, first proposed in (Donoho et al., 2012) is termed stagewise selection. In MP and OMP, the selection step chooses one element that is maximally correlated with the residual. A very natural stagewise strategy is to replace the maximum with a threshold criterion and select more than one element per iteration. In the Stagewise Orthogonal Matching Pursuit (StOMP) described by (Donoho et al., 2012), a threshold $\lambda^{[i]}$ is defined as a function of the residual $\mathbf{r}^{[i-1]}$. A specific problem is that the algorithm terminates prematurely when all inner products fall below the threshold.

Another alternative multi-element selection strategy that has been proposed is the Regularized OMP (ROMP) (Needell and Vershynin, 2008; Needell and Vershynin, 2010), which groups the inner products $\mathbf{g}_i$ into sets J_k such that the elements in each set have similar magnitude. ROMP then selects the set J_k for which $\sum_{j \in J_k}(|\mathbf{g}_i|)^2$ is largest.

Thresholding type methods

As shown above, greedy pursuits are easy to implement. Moreover, they can be extremely fast. However, they do not have recovery guarantees that are as strong as methods based on convex relaxation. The methods discussed here bridge this gap. They are relatively easy to implement and can be extremely fast and show strong performance guarantees with convex relaxation strategies.

The iterative hard thresholding algorithm is introduced by Blumensath and Davies (2008b). It is a greedy algorithm that iteratively solves a local approximation to the CS recovery problem. At each iteration, the algorithm greedily finds a global minimum based on the current estimate $\hat{\mathbf{x}}^{[i]}$,

$$\hat{\mathbf{x}}^{[i+1]} = H_k(\hat{\mathbf{x}}^{[i]} + \mu \mathbf{A}^T(\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}^{[i]})), \tag{1.6}$$

where H_k is the hard thresholding operator that sets all but the largest k elements of its argument to zero. The IHT algorithm is easy to implement and is computational efficient.

There are also some other greedy methods like Compressive Sampling Matching Pursuit (CoSaMP) algorithm by Needell and Tropp (2009), and the Subspace Pursuit (SP) algorithm by Dai and Milenkovic (2009).

1.3 Robust sparse recovery

Compressive Sensing and sparse reconstruction methods generally apply to noise-free signals or signal contaminated with random noise, preferable with Gaussian distribution. However, geophysical datasets are often corrupted by erratic noise of unknown distribution. It is well known that least squares-based estimators are highly sensitive to erratic noise in the observed data, leading to poor performance when the noise does not follow the Gaussian assumption and is, instead, better characterized by heavier-than Gaussian-tailed distributions (Kassam and Poor, 1985; Zoubir et al., 2012).

Therefore, in Chapter 2, I will discuss a robust theory for sparse reconstruction, and I will also introduce some well-accepted algorithms for robust sparse reconstruction. These algorithms will become the engine adopted in this thesis for simultaneous source seismic data processing. Simultaneous-source seismic data processing offers an excellent scenario for testing denoising methods that require robust sparse approximations.

1.3.1 Simultaneous source separation

Simultaneous source acquisition techniques have gained popularity as a low-cost strategy to improve seismic data acquisition (Beasley, 2008; Berkhout, 2008). In the simultaneous source acquisition configuration, instead of firing one shot and imposing large time intervals between different shots, several seismic sources fire at close time intervals. The same set of receivers then records the response of multiple sources. By allowing overlaps between the closely fired shots, we can acquire various shot records during the period that one could have used to obtain one conventional shot record. The main benefit of simultaneous source acquisition is that one can significantly increase source density and improve seismic images with extra acquisition cost. Figure 1.2 shows an example of simultaneous source acquisition with two sources towed by two vessels. Two sources move in opposite directions and fire sources with a small time difference. The primary technical challenge for simultaneous source seismic data processing lies in the strong interference introduced by the closely fired shots. Therefore, an additional processing step referred to as simultaneous source separation or deblending must apply to the conventional processing flow. The goal is to separate the responses from each shot and to eliminate simultaneous source interferences. After deblending or separation, the data should be comparable to the ideal unblended data that one could have acquired from the conventional seismic acquisition.

The pseudo-deblending process shifts the firing time delay back for each shot. As shown in the figure 1.3, the desired signal appears coherent in common receiver, common offset, and common midpoint domains of the pseudo-deblended data. In contrast, interferences can be modelled as erratic noise. The interferences from the blended shots would appear random because the randomized firing time delays perturb them. Meanwhile, both desired signals and interferences are coherent in the common shot gather. Therefore, the deblending problem can be solved by removing erratic noise in the common receiver, common offset, and common midpoint gathers, which can be done by the robust sparse reconstruction methods. In essence, deblending in this thesis is tackled as a robust denoising problem.

1.4 Contributions of this thesis

My thesis is about robust Compressive Sensing and robust sparse recovery algorithms and mainly focused on robust greedy methods and their application to simultaneous source data processing. My application is to seismic data processing but the algorithms developed in this thesis can be applied to problems in other areas.

- I first introduce the general concepts of compressive Sensing and sparse recovery. I discuss two categories of traditional recovery algorithms, which are

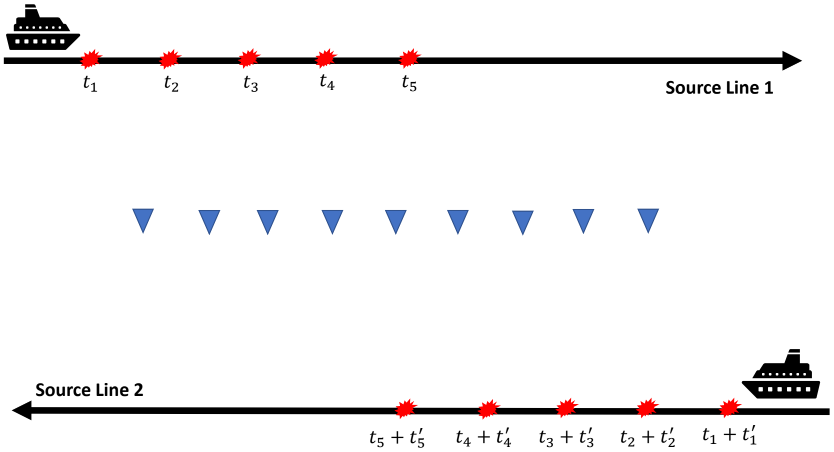

Figure 1.2: Simultaneous source acquisition with two sources. Red symbols represent sources. Blue triangles represent receivers. In a conventional acquisition each time one source is activated the array of receivers only record the response for that particular source. In simultaneous source acquisition, receivers record the responses of two (or more) sources activated with random time delays. The process of deblending or separation involves retrieving the responses from individual sources from simultaneous source data.

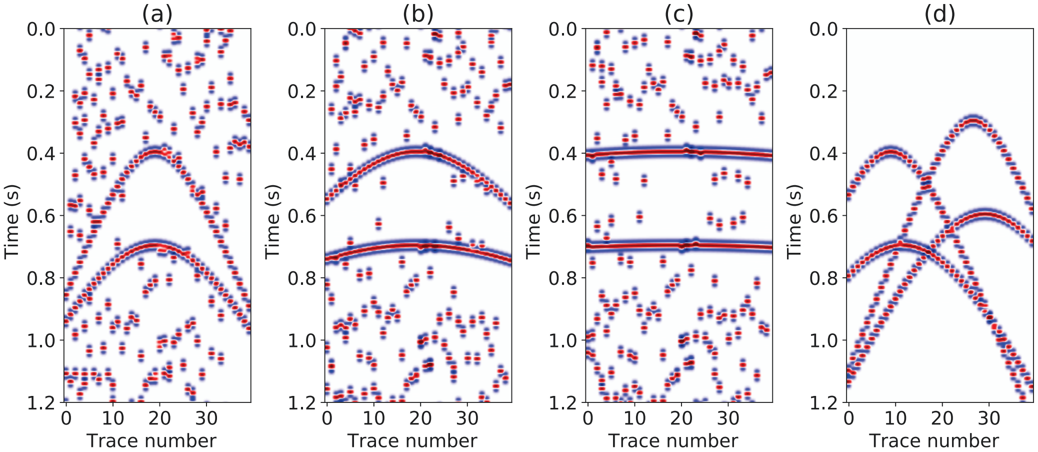

Figure 1.3: Data after pseudo-deblending. (a) A common midpoint gather. (b) A common receiver gather. (c) A common offset gather. (d) common shot gather. In a common shot gather the responses of the two sources are coherent and therefore, we cannot apply separation algorithms based on denoising in this domain. On the other hand, in all the other domains, the response for one source is coherent and incoherent for the rest of the sources making these domains attractive for source separation techniques based on denoising.

convex relaxation and greedy methods. Then I review the theory of the robust sparse reconstruction and many robust sparse reconstruction algorithms. I also take some simple tests to compare the performance of the robust and non-robust reconstruction algorithms.

- I present the classical Matching Pursuit algorithm (Mallat and Zhang, 1993) using the simple language of linear algebra, and then I make it robust to the presence of outliners in the data. The modification entails replacing the ℓ_2 space inner product required to identify basis functions by a ℓ_p space inner product. The ℓ_p space inner product helps determine the correct basis function in each iteration of the Matching Pursuit scheme.
- My contribution also addresses technical questions concerning Matching Pursuit implementation for cases where the basis functions are not given in an explicit form. An example of the latter is the time domain Radon transform adopted in this study to denoise simultaneous source records.
- The traditional Matching Pursuit algorithm is relatively slow when is applied to complex synthetic data and real marine data. Therefore, I also discuss other greedy pursuit algorithms that works much faster than the classical MP algorithm.

1.5 Thesis Outline

The structure of the thesis is as follow

- In **Chapter 1**, I provide an introduction about Compressive Sensing and sparse reconstruction algorithms. These algorithms can be divided into two groups, which are convex relaxation methods and greedy methods. I then discuss the challenge we faced with traditional sparse recovery algorithms. These algorithms only work when the signals contain no noise or have random noise that follows a normal distribution. This problem motivates the thesis core topic: Robust sparse recovery when the signal contains impulsive or erratic noise.

- In **Chapter 2**, I review the theory of robust reconstruction. I also introduce some commonly used robust M-estimators. I review algorithms to solve the robust sparse reconstruction problem using different M-estimators. I consider simple tests to compare the performance between robust and non-robust sparse reconstruction algorithms.

- In **Chapter 3**, I introduce a new robust greedy method called $lp - norm$ matching pursuit. It replaces the $l2 - norm$ inner product used in the traditional matching pursuit algorithm with our new-defined robust inner product. I also explain the implementation of Matching Pursuit with an operator given in the implicit form, in our case, a local Radon transform. Finally, I apply the robust Matching Pursuit with local Radon transform to the simultaneous source separation problem.

- In **Chapter 4**, I discuss other greedy pursuit algorithms that work faster than the traditional MP algorithm. Then I explained how to apply robust Radon transform and robust inner product into these faster greedy pursuit algorithms. I also provide tests to compare the performance of each algorithm in the different scenarios.

- In **Chapter 5**, I conclude the main results and the observations of my research. In the end, I discuss the limitations of the proposed approach and provide suggestions for future works.

CHAPTER 2

Robust sparse inversion

In Chapter 1, I provide an introduction to compressive sensing (CS) and sparse reconstruction algorithms. Traditional techniques for CS and sparse representation use the ℓ_2 norm as the metric for the residual error. However, it is well known that least-squares based estimators are highly sensitive to outliers present in the data. The performance of least-squares methods is only guaranteed when the noise follows the Gaussian assumption. The presence of noise that follows a heavy-tailed distribution will lead to poor performance. In the area of geophysical signal processing, often data contain heavy-tailed noise also called erratic noise. One way of addressing this type of noise is via robust statistics and, more precisely, by robust filtering techniques.

2.1 M-estimators

Robust signal processing techniques are generally based on maximum likelihood type estimators. Maximum likelihood (ML) type estimators, also known as M-estimators, are described by a cost function-defined optimization problem where properties of the cost function (or its first derivative, the so-called influence function) determine the estimator robustness (Hampel et al., 1986). An estimator's robustness is characterized by two essential concepts: the breakdown point and the influence function. The breakdown point is used to describe the quantitative robustness of an estimator. It indicates the maximal fraction of outliers in the observations, which an estimator can handle without breaking down. The influence function describes the bias impact of infinitesimal contamination at an arbitrary point on the estimator, standardized by the fraction of contamination. For M-estimators, the influence function is proportional to the first derivative of the cost function(Hampel et al., 1986).

Let us assume a linear model where observations $\mathbf{y}$ are related to model parameters or coefficients $\mathbf{x}$ via a linear operator or matrix $\mathbf{A}$. Under the above mentioned model, observations can be written as follows

$$\mathbf{y} \approx \mathbf{A}\mathbf{x} \tag{2.1}$$

where the symbol $\approx$ emphasizes that we have noise in the system and therefore, $\mathbf{A}\mathbf{x}$ is understood as modelled data that should not fit the observation exactly. In sparse approximation theory or sparse inverse problems our goal is to find a sparse solution of the unknown $\mathbf{x}$ and therefore, we pose the problem as one where we minimize a cost function given by

$$J = \|\mathbf{r}\|_2^2 + \lambda \|\mathbf{x}\|_1^1 \,. \tag{2.2}$$

The ℓ_1 term is the sparsity promoting term. The vector $\mathbf{r}$ are the residuals $\mathbf{r} = \mathbf{y} - \mathbf{A}\mathbf{x}$ and λ is the tradeoff parameter. Notice that in equation 2.2 I have adopted an ℓ_2 norm for the residuals. The latter is equivalent to assume a Gaussian distribution of errors. Therefore, if the data is contaminated by erratic noise, the solution $\mathbf{x}$

will contain unreal features that are needed to explain outliers. This effect can be attenuated by introducing a robust M-estimator. In others words, the main idea is to adopt a different manner of measuring error. In the ℓ_2 case, the "size of the residuals" is measured by

$$\|\mathbf{r}\|_2^2 = \sum_i r_i^2 = \sum_i f(r_i) \tag{2.3}$$

where it is clear that $f(x) = x^2$ is the function used when we adopt the ℓ_2 norm [1]. The M-estimator tries to reduce the effect of outliers by replacing $f(\cdot)$ by a function $\rho(\cdot)$ which is robust to outliers. Then, a typical sparse inversion problem with data contaminated by erratic noise becomes one where we minimize the following cost function

$$J = \|\mathbf{r}\|_\rho + \lambda \|\mathbf{x}\|_1^1 \,, \tag{2.4}$$

where

$$\|\mathbf{r}\|_\rho = \sum_i \rho(r_i) \tag{2.5}$$

Generally, this problem can be solved by the Iterative Reweighted Least-Squares method (IRLS) (Holland and Welsch, 1977; Ji, 2006; Ibrahim and Sacchi, 2014) with the following equivalent cost function

$$J = \|\mathbf{W_r r}\|_2^2 + \lambda \|\mathbf{W_x x}\|_2^2. \tag{2.6}$$

Thus, we have turned the non-quadratic problem into a sequence of quadratic minimization problems where $\mathbf{W_r}$ and $\mathbf{W_x}$ are diagonal matrices that depend on the residual $\mathbf{r}$ and model $\mathbf{x}$, respectively. These matrices are calculated from the solution $\mathbf{r}$ and $\mathbf{x}$ in an iterative manner. The function ρ, its derivative (also called the

[1] or $f(x) = |x|^2 = xx^*$ for complex x

M-estimators			
type	$\rho(x)$	$\psi(x)$	$w(x)$
ℓ_2	$\frac{x^2}{x}$	x	1
ℓ_1	$\|x\|$	$sgn(x)$	$\frac{1}{\|x\|}$
$\ell_1 - \ell_2$	$2(\sqrt{(1+x^2/2)}-1)$	$\frac{x}{\sqrt{(1+x^2/2)}}$	$\frac{1}{\sqrt{1+x^2/2}}$
ℓ_p	$\frac{\|x\|^p}{p}$	$sgn(x)\|x\|^{p-1}$	$\|x\|^{p-2}$
Huber $\begin{cases} if\|x\| \leq k \\ if\|x\| \geq k \end{cases}$	$\begin{cases} x^2/2 \\ k(\|x\| - k/2) \end{cases}$	$\begin{cases} x \\ ksgn(x) \end{cases}$	$\begin{cases} 1 \\ k/\|x\| \end{cases}$
Cauchy	$\frac{c^2}{2}log(1+(x/c)^2)$	$\frac{x}{1+(x/c)^2}$	$\frac{1}{1+(x/c)^2}$
Geman-McClure	$\frac{x^2/2}{1+x^2}$	$\frac{x}{(1+x^2)^2}$	$\frac{1}{(1+x^2)^2}$
Welsch	$\frac{c^2}{x}[1-exp(-(x/c)^2)]$	$xexp(-(x/c)^2)$	$exp(-(x/c)^2)$
Tukey $\begin{cases} if\\|x\\| \leq c \\ if\\|x\\| \geq c \end{cases}$	$\begin{cases} \frac{c^2}{6}(1-[1-(x/c)^2]^3) \\ c^2/6 \end{cases}$	$\begin{cases} x[1-(x/c)^2]^2 \\ 0 \end{cases}$	$\begin{cases} [1-(x/c)^2]^2 \\ 0 \end{cases}$

Table 2.1: A few commonly used M-estimators.The variable x can be either an element of the residual vector $\mathbf{r}$ or an element of the vector of model parameters $\mathbf{x}$. In compressive or sparse inversion problems, we often select the ℓ_1 norm for the regularization term and the ℓ_2 norm for the error term. In robust sparse inversion or robust compressive sensing, we will continue to adopt the ℓ_1 norm for the regularization term and adopt one of the M-estimators above (except for the ℓ_2 norm which is non-robust.)

influence function) and the expression of the weights for different M-estimators are listed in table 2.1. The function $w(x)$ or $w(r)$ are used to compute the diagonal matrices of weights as follow

$$[\mathbf{W_r}]_{ii} = w(r_i) \tag{2.7}$$

and

$$[\mathbf{W_x}]_{ii} = w(x_i)\,. \tag{2.8}$$

Table 2.1 shows some commonly use M-estimators, $\rho(x)$ is the functional that affects the error fitting term. Similarly, $\psi(x)$ is the corresponding influence function which is proportional to the first derivative of $\rho(x)$, and $w(x)$ is the weighting matrix that we used to solve the problem by, for instance, the iterative reweighted least square (IRLS) method.

Figure 2.1,2.2 and 2.3 show the cost function, influence function and weighting function of the M-estimators, respectively. As we can see from theses figures, the ℓ_2 norm estimator is non-robust because its influence function is not bounded, as the $\mathbf{x}$ increases, the influence function also increases linearly. The ℓ_1 estimator reduce the influence of the large errors, but they still have some influence because the influence function has no cut-off point. In addition, the ℓ_1 estimator is not stable because the ρ function is convex but its derivative does not exist at at $x = 0$. The latter implies that special care needs to be taken to deal with the discontinuity of gradients at $x = 0$. The $\ell_1 - \ell_2$ estimator can reduce the influence of large errors like ℓ_1, and is also convex. The influence of the Huber estimator has the similar behaviour as the ℓ_1 estimator. For the Cauchy[2], Geman-McClure and Welsh estimators, the influence of large errors only decreases linearly. For the Tukey function, the outliers are totally suppressed. As we mentioned before, these problems can be generally solved by IRLS. Ji (2006) explains how to solve the robust inversion problem by IRLS in conjunction with the conjugate gradient method in detail.

2.2 Methods

We first consider the ℓ_1 norm case where one seeks a robust method to estimate a sparse solution

$$\underset{\mathbf{x}}{min} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_1^1 + \lambda\|\mathbf{x}\|_1, \tag{2.9}$$

[2] Also known as the Lorentzian function.

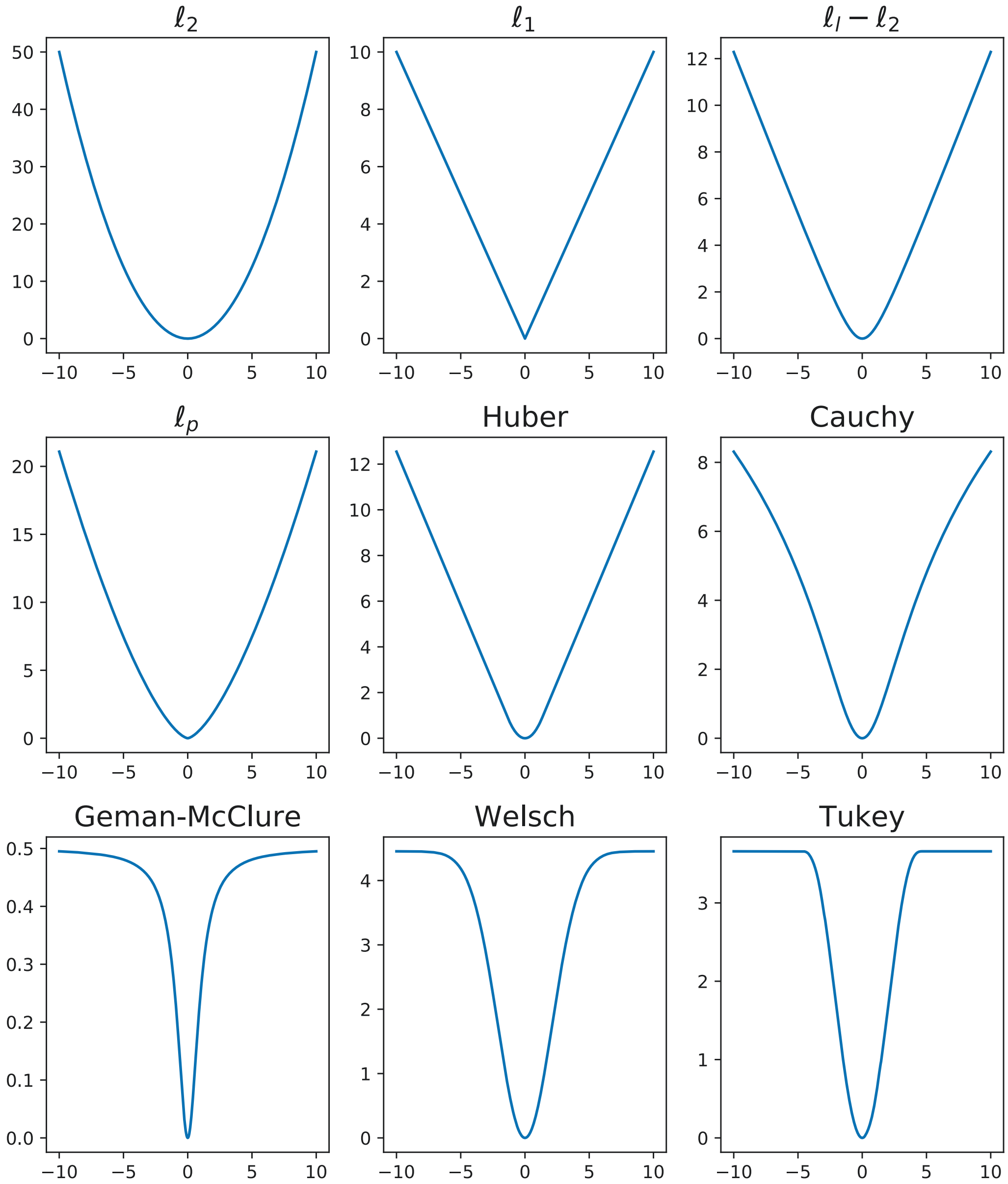

Figure 2.1: Cost function of different M-estimators. For the ℓ_p norm we use $p = 1.5$, and for Huber estimator we use $k = 0.75$. we use $c = 1$ for the Cauchy, Welsch and Tukey norms.

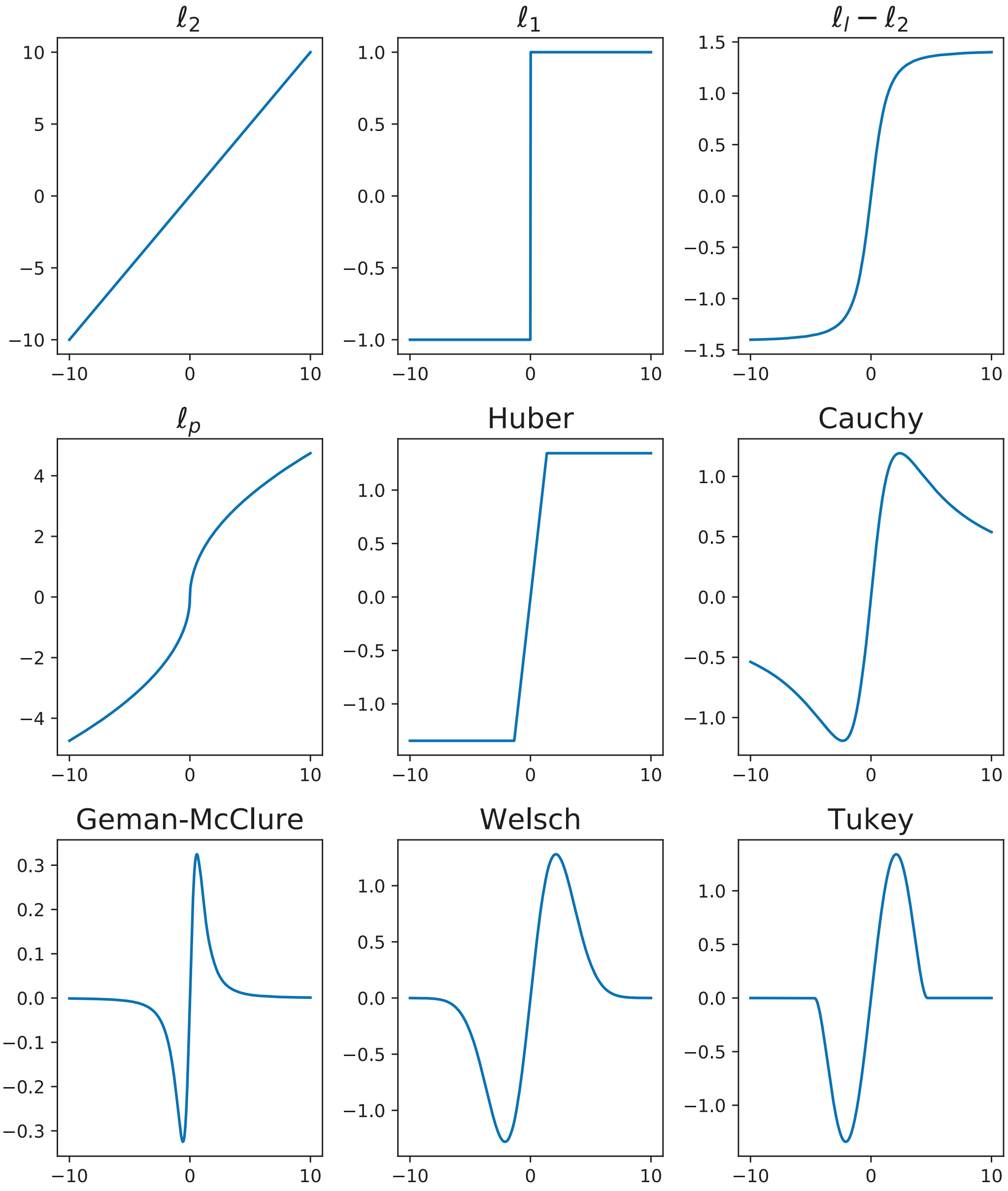

Figure 2.2: Influence function of different M-estimators. For the ℓ_p norm we use p=1.5. For the Huber estimator we use $k = 0.75$. We use $c = 1$ for the Cauchy, Welsch and Tukey estimators.

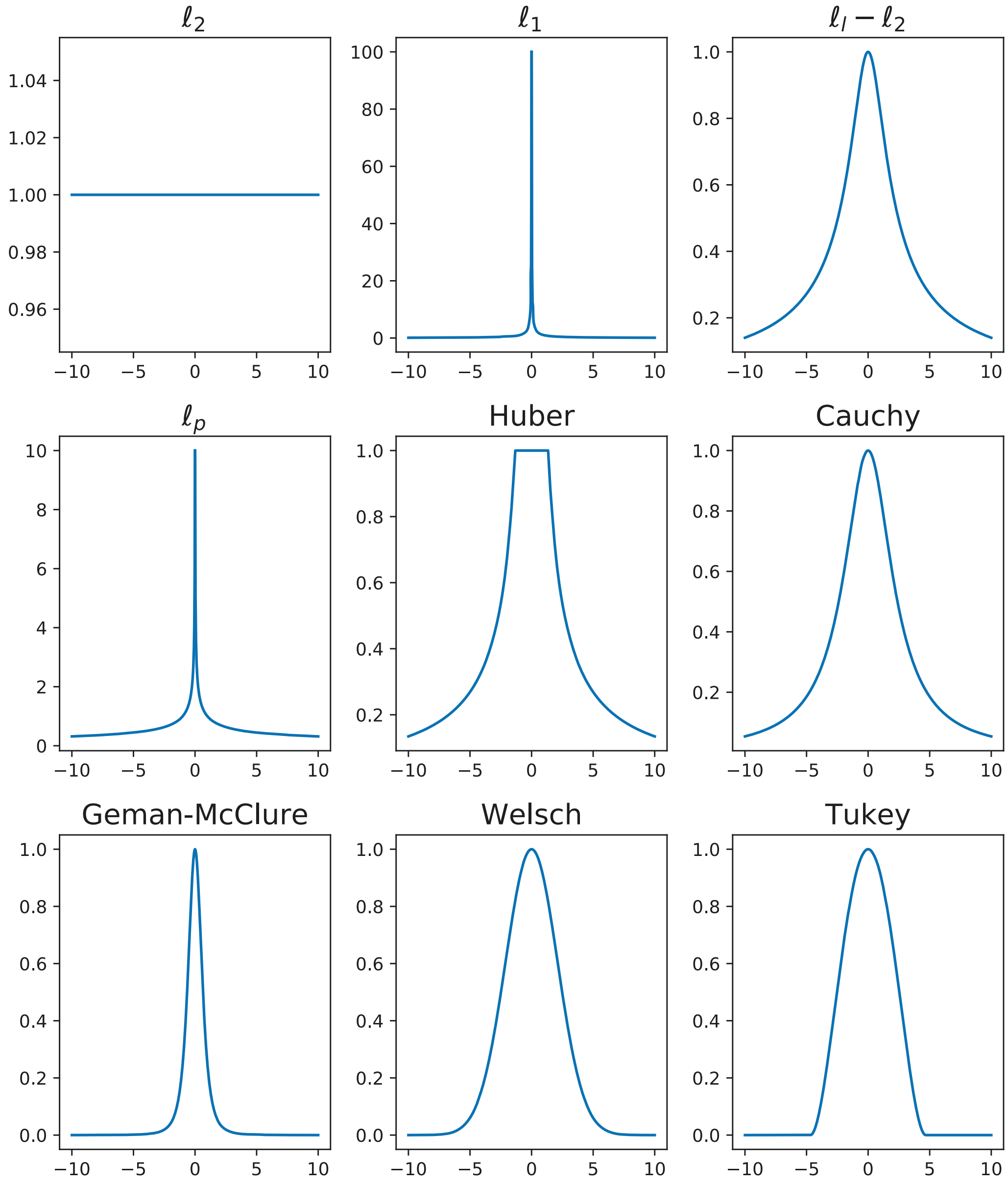

Figure 2.3: Weighting function for different M-estimators. For the ℓ_p norm we use p=1.5. For the Huber estimator we use $k = 0.75$. We adopted $c = 1$ for the Cauchy, Welsch and Tukey estimators.

Equation 2.9 is called a LAD l_1 regularized problem (l_1-LAD) and has been studied by Wang et al. (2007). This problem can be changed to the following equivalent convex problem

$$\underset{\mathbf{x},\mathbf{z}}{min} \quad \lambda\|\mathbf{x}\|_1 + \|\mathbf{z}\|_1 \quad \text{subject to} \quad \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z} \tag{2.10}$$

where $\mathbf{z}$ represents the erratic noise. The parameter λ is the trade-off parameter controls the two l_1 terms. In signal processing, this method is often referred as Justice Pursuit (Laska et al., 2009). The theoretical recovery conditions for solving this problem have been studied by Wright and Ma (2009); Studer et al. (2011); Li (2012). Equation 2.10 can also be modified as follows for the case of data contaminated with both erratic noise and non-erratic (Gaussian) noise

$$\underset{\mathbf{x},\mathbf{z}}{min} \lambda\|\mathbf{x}\|_1 + \|\mathbf{z}\|_1 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x} - \mathbf{z}\|_2 \leq \epsilon \,. \tag{2.11}$$

The theoretical recovery conditions for the problem above have been studied by Nguyen and Tran (2013); Nguyen and Tran (2011); Studer and Baraniuk (2013). The problem posted in 2.11 is convex and can be efficiently solved by using an optimization algorithm like FISTA (Beck and Teboulle, 2009).

The problem in 2.10 is convex but non-smooth, it can be solved using the alternating direction method of multipliers (ADMM) (Yang and Zhang, 2011).

If the erratic noise has heavier tails than those of a Laplace distribution (ℓ_1 norm case), we can use the ℓ_p norm with $0 < p < 1$. Then, the problem becomes

$$\underset{\mathbf{x}}{min} \, \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_p^p + \lambda\|\mathbf{x}\|_1 \,. \tag{2.12}$$

An algorithm that uses the proximity operator of the ℓ_p norm in conjunction with

ADMM has been proposed to efficiently solve 2.12 for the case $0 < p < 2$ by Wen et al. (2016). For the non-convex case $0 < p < 1$, a smoothing strategy has also been proposed to derive a convergent algorithm.

Another method to solve 2.12 is a greedy method. For instance, Zeng et al. (2016) proposed a robust MP and OMP method called ℓ_p-MP and ℓ_p-OMP, respectively. Both techniques are based on the ℓ_p-space correlation defined as follows

$$c_p(\mathbf{a}, \mathbf{b}) = 1 - \frac{min_\alpha \|\mathbf{b} - \alpha\mathbf{a}\|_p^p}{\|\mathbf{b}\|_p^p}. \tag{2.13}$$

One can also use the Cauchy cost function to replace the ℓ_2 norm for the data-fitting term. In this case, the problem becomes

$$\underset{\mathbf{x}}{min} \sum_{i=1}^{m} \rho(\frac{y_i - A_i^T\mathbf{x}}{\sigma}) + \lambda\|\mathbf{x}\|_1, \tag{2.14}$$

with $\rho(\cdot)$ the functional corresponding to the Cauchy norm (Table 2.1). The parameter σ is the scale parameter of the Cauchy distribution The scalar λ is a regularization parameter that controls the sparsity level of the solution. Equation 2.14 can be solved efficiently based on the fast iterative shrinkage algorithm (FISTA) (Pham and Venkatesh, 2012) and ADMM (Pham and Venkatesh, 2013).

2.2.1 Denoising by robust sparse inversion

At this point I have described methods to estimate sparse solutions for linear problems. Now, I would like to make the connection to denoising, the central topic of my thesis. Problems defined in the previous section fall under the general scheme

$$\widehat{\mathbf{x}}_\lambda = \underset{\mathbf{x}}{\text{argmax}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_\rho + \lambda\|\mathbf{x}\|_1 \tag{2.15}$$

where it is clear that $\widehat{\mathbf{x}}_\lambda$ is the solution of our problem for a suitable parameter λ. The solution $\widehat{\mathbf{x}}_\lambda$ can be used to predict the data

$$\widehat{\mathbf{y}} = \mathbf{A}\widehat{\mathbf{x}}_\lambda \,. \tag{2.16}$$

We call $\widehat{\mathbf{y}}$ the predicted or denoised data. In general, $\mathbf{A}$ is a synthesis operator that transforms coefficients $\mathbf{x}$ to data $\mathbf{y}$. The operator $\mathbf{A}$ can be the inverse Fourier transform in which case, $\mathbf{x}$ is the vector of complex Fourier coefficients (Sacchi et al., 1998). Similarly, $\mathbf{A}$ can be a Radon operator that maps Radon-domain coefficients to data space (Sacchi and Ulrych, 1995). In general, when we refer to denoising we mean to estimate $\mathbf{x}$ from $\mathbf{y}$ and then use the result to synthesize $\widehat{\mathbf{y}}$ via equations 2.15 and 2.16.

2.3 Deblending by robust sparse inversion

The simultaneous source separation problem can be tacked via robust denoising. In Chapter 1, I have also presented robust sparse inversion via robust M-estimators. In this section, I test the performance of these methods with an application that entails denoising via the robust Radon transform.

2.3.1 Radon transform

One of the basic principles for Compressive sensing (CS) and sparse inversion is that the signal is sparse under a particular transform. The seismic noise removal methods are based on the principle that the seismic data can be transfered into a new domain where the signal and noise can be separated (Berkhout and Verschuur, 2006). The Radon transform is one of these suitable transformations because it can focus seismic signals efficiently (Deans, 2007; Kuchment, 2013). It has been well used in seismic data processing applications, such as interpolation (Kabir and Verschuur, 1995; Sacchi and Ulrych, 1995; Trad et al., 2002), multiple separation (Hampson, 1986a; Trad et al., 2003a) and noise removal (Russell et al., 1990a,b). Many deblending methods are also based on Radon transform (Moore et al., 2008; Akerberg et al., 2008; Ibrahim and Sacchi, 2014).

The forward Radon operator is denoted by L and the adjoint by L^*. Having these two operators is equivalent to having access to the matrix $\mathbf{A}$ (or its columns $\mathbf{a}_i$) and its Hermitian transpose $\mathbf{A}^H$ in the preceding section. For a general Radon transform with integration path $t = \tau + q\phi(h)$ the forward and adjoint operators are given by

$$L : d(t,h) = \sum_q a(\tau = t - q\,\phi(h), q), \tag{2.17}$$

$$L^* : \tilde{a}(\tau,q) = \sum_h d(t = \tau + \phi(h)q, h), \tag{2.18}$$

where h is offset, t is time and τ is intercept. For a linear Radon transform, $\phi(h) = h$ and q represents dip or ray-parameter. Similarly, for the parabolic Radon transform, $\phi(h) = (h/h_f)^2$ and q represents residual parabolic moveout at far offset h_f (Hampson, 1986b). The Radon coefficients are given by $a(\tau,q)$ and the low-resolution coefficients computed via the adjoint operator are given by $\tilde{a}(\tau,q)$. In our numerical implementation it is clear that all variables are discrete and, for instance, h is replaced by the discrete offset $h_i, i = 1 \dots n_h$. Similarly, q is replaced by the discrete Radon parameter $q_i, i = 1 \dots n_q$.

2.3.2 Blending operator

The next important concept is the blending operator, which I use to create data that mimics a simultaneous acquisition survey. First, I assume that the seismic data acquired via a conventional seismic acquisition survey are denoted as $D(t, r, s)$, where t, r, s are used to indicate the time, receiver, and source indices, respectively. In simultaneous source acquisition, the trace recorded by the j-th receiver (r_j) can be simulated as

$$b(t, r_j) = \sum_{i \in S} D_{r_j}(t - \tau_i, r_j, s_i), \tag{2.19}$$

where S indicates a group of shots with fire times and locations (τ_i, s_i). This equation can be rewritten into operator form like

$$\mathbf{b} = \mathbf{BD}, \tag{2.20}$$

where $\mathbf{B}$ is the blending operator defined by Berkhout (2008) and $\mathbf{b}$ are the blended data. The symbol $\mathbf{D}$ is the ideal unblended common receiver gather for the receiver j. Basically, the blending operator shifts each shot record according to the fire time and then superposes the shot records into a blended shot gather. The adjoint operator, of the blending operator, is the pseudo-deblending operator Berkhout (2008). We denote the pseudo-deblending operator as follows

$$\hat{\mathbf{D}} = \mathbf{B}^{*}\mathbf{b}. \tag{2.21}$$

Pseudo-deblending entails shifting the fire time delays back to every single shot and then truncating the blended shot record to the conventional shot's recording interval. Figure 2.4 and 2.5 illustrate the numerical blend and pseudo-deblend the three shots. As we can see from figure 2.5, the pseudo-deblend process can not

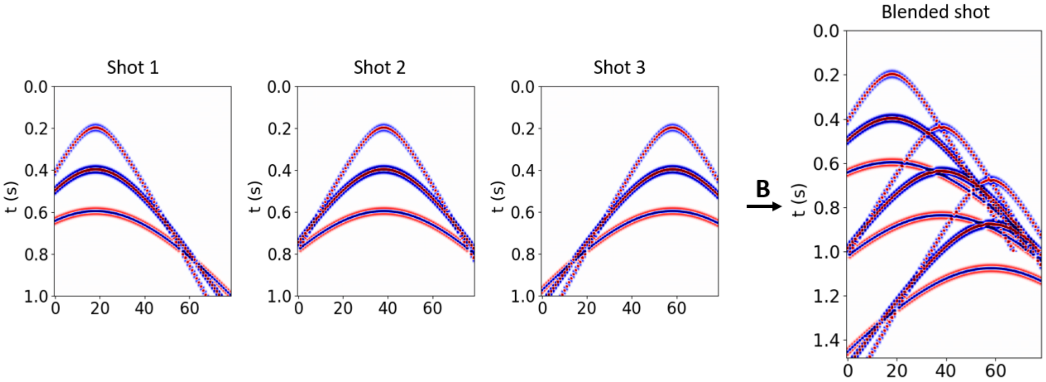

Figure 2.4: Illustration of numerical blending using three shots.

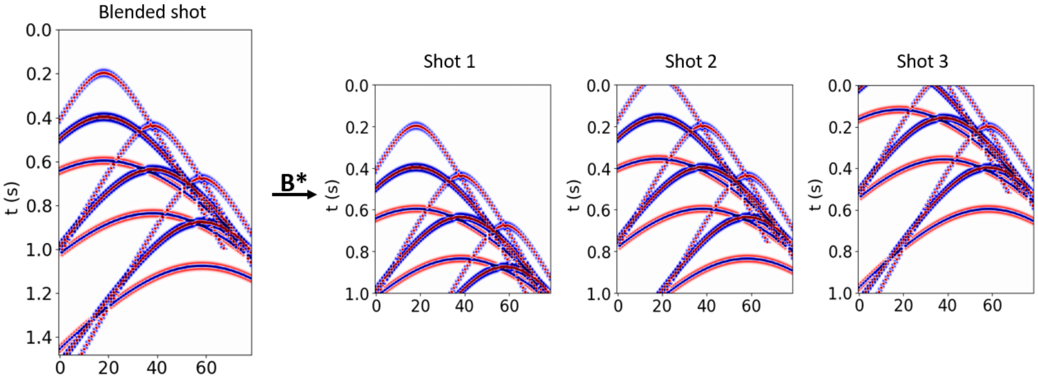

Figure 2.5: Illustration of pseudo-deblending of two shots.

remove the source interference.

2.3.3 Blending factor

The blending factor is defined as the average number of simultaneous source shots that fit the desired conventional record. A blending factor equals five means we fire five sources fired on one blended record. This means the blending factor for the example in figure 2.4 is three. With this definition, a high blending factor indicates a higher level of blending noise.

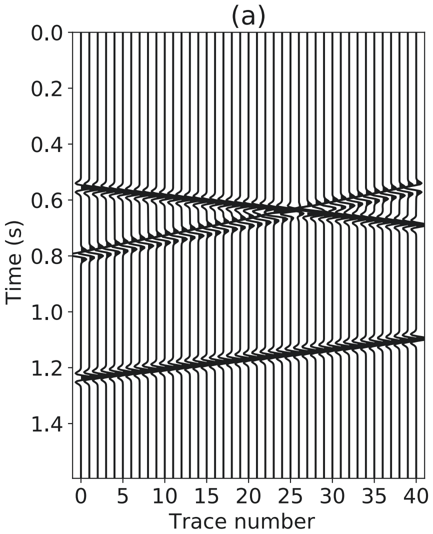

Figure 2.6: Simple synthetic data generated to test robust and sparse denoising algorithms based on the Radon transform.

2.3.4 Examples

The goal of the test is to estimate the Radon coefficients that model the data and then use them to predict the denoised data. I will measure the quality of the reconstruction via the following expression

$$SNR = 10\log_{10}(\frac{\|\mathbf{y}^c\|_2^2}{\|\mathbf{y}^c - \widehat{\mathbf{y}}\|_2^2})$$

where $\mathbf{y}^c$ is the clean data. In all our examples, we add erratic noise to $\mathbf{y}^c$ by the blending and pseudo-deblending operators, and trying to mimic blending noise. Then, we estimate $\widehat{\mathbf{y}}$ via robust ℓ_p and non-robust ℓ_2 sparse inversions. The noisy data with strong erratic noise can be considered as the pseudo-deblended common receiver gathers. Figure 2.6 to figure 2.10 shows the denoising results when we use different value of the parameter p for the ℓ_p norm estimator. We test the performance with three different noise levels defined by the different blending factors for each one.

Figure 2.11 shows the denoising performance changes as the value of p. From these figures, we can see that, as the value of p decreases, our algorithm becomes more robust to blending noise. However, the algorithm's performance is susceptible to the value of the tradeoff parameter λ. As we can see from figures 2.12 to figure 2.15, a small change of λ can change the final performance of the denoiser. For the same cost function, data with a different noise level needs to use a different value of λ. Similarly, for the same dataset, we need to find the best λ when we use a different value of the parameter p in the cost function. There is no clear method to calculate this λ, especially when dealing with non-Gaussian noise data. A heuristic approach is to monitor the solution (predicted denoised data) and visually inspect the results for evidence of over or under-fitting and modify the tradeoff parameter accordingly. We run these tests by both IRLS and ADMM methods. The performance of these algorithms is similar. The final results of both methods are susceptible to the value of λ. These techniques are challenging to apply to real datasets methods. In the next chapters, I propose to adopt robust greedy methods for denoising blended datasets similar to the data used in this section.

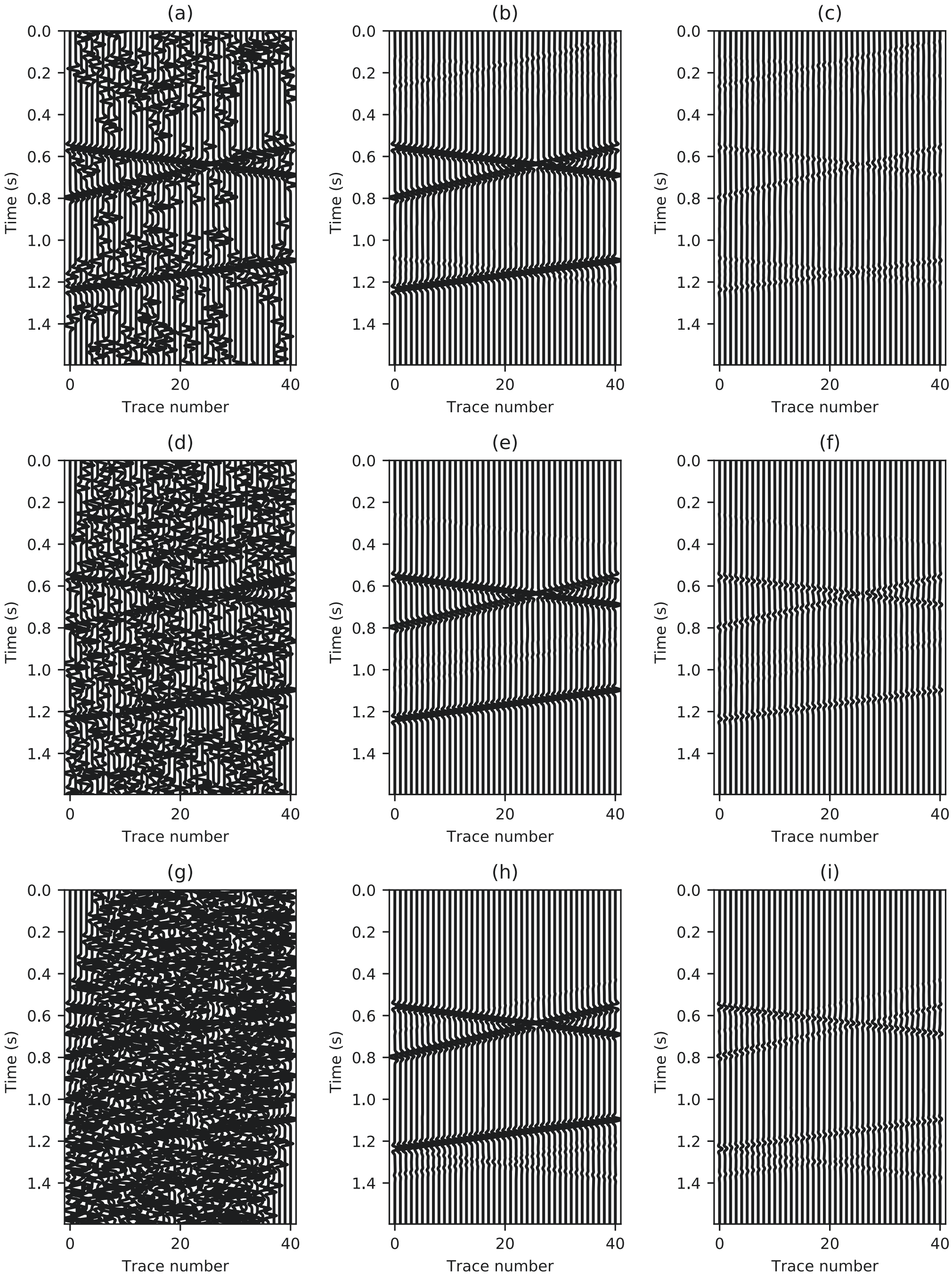

Figure 2.7: Denoising results for the error ℓ_2 estimator (non-robust solution). (a) Noisy data with a blending factor of 5. (b) The denoised data yield $SNR = 15.5$ dB. (c) Errors between the denoised data and the clear data in Figure 2.6. (d) Noisy data with a blending factor of 10. (e) Denoised data $SNR = 13.1$ dB. (f) Errors between e and the clean data. (g) Noisy data with a blending factor of 20. (h) Denoised result SNR=12.2 dB. (i) Errors between h and the clear data.

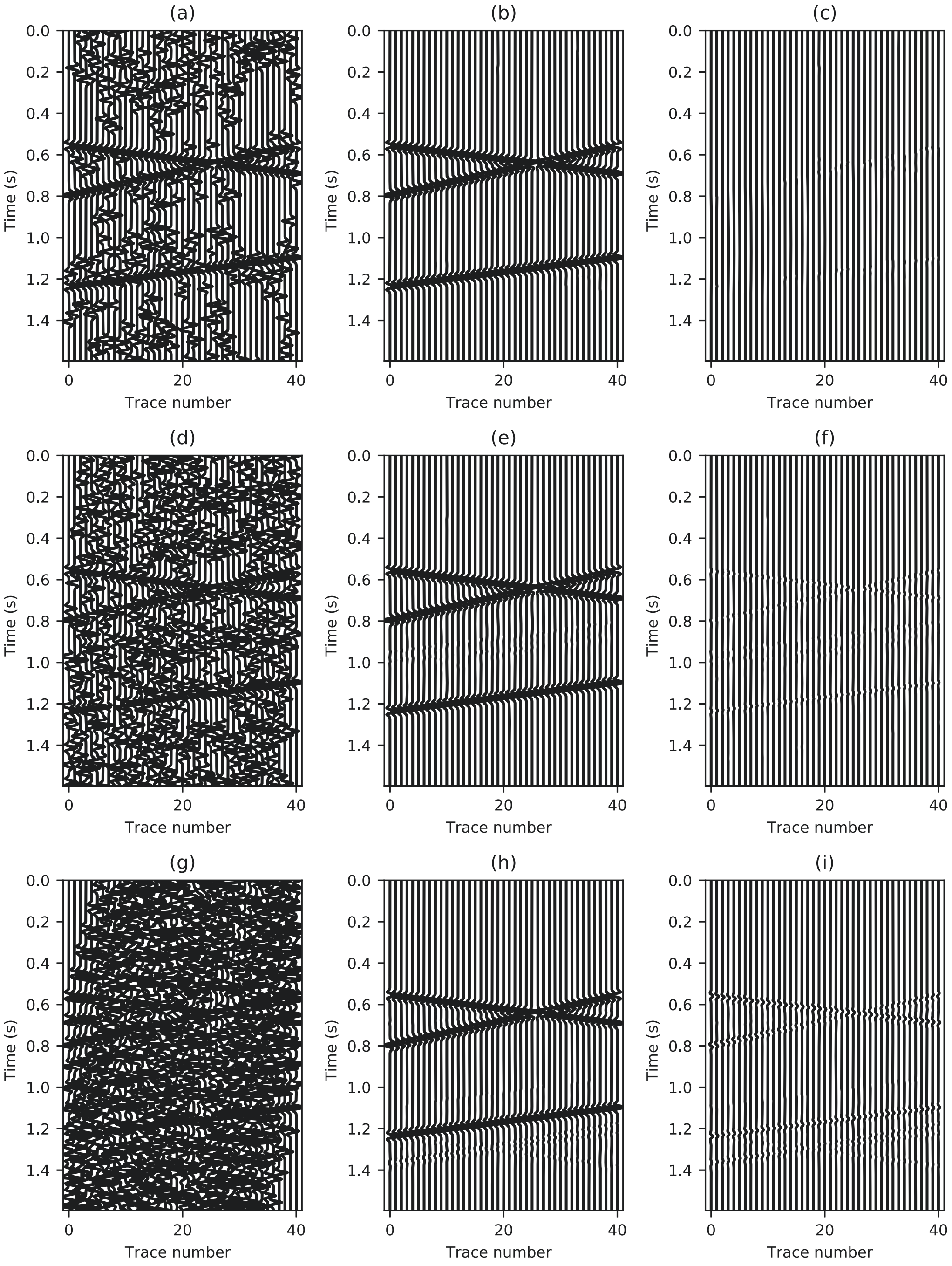

Figure 2.8: Denoising results for the ℓ_p estimator with $p = 1.5$. (a) Noisy data with a blending factor of 5. (b) Denoised data result $SNR = 28.4$ dB. (c) Errors between b and the clean data. (d) Noisy data with a blending factor of 10. (e) Denoised data $SNR = 21.8dB$. (f) Errors between e and the clean data (g) Noisy data with a blending factor of 20. (h) denoised data $SNR = 12.2db$. (i) Difference between h and the clean data.

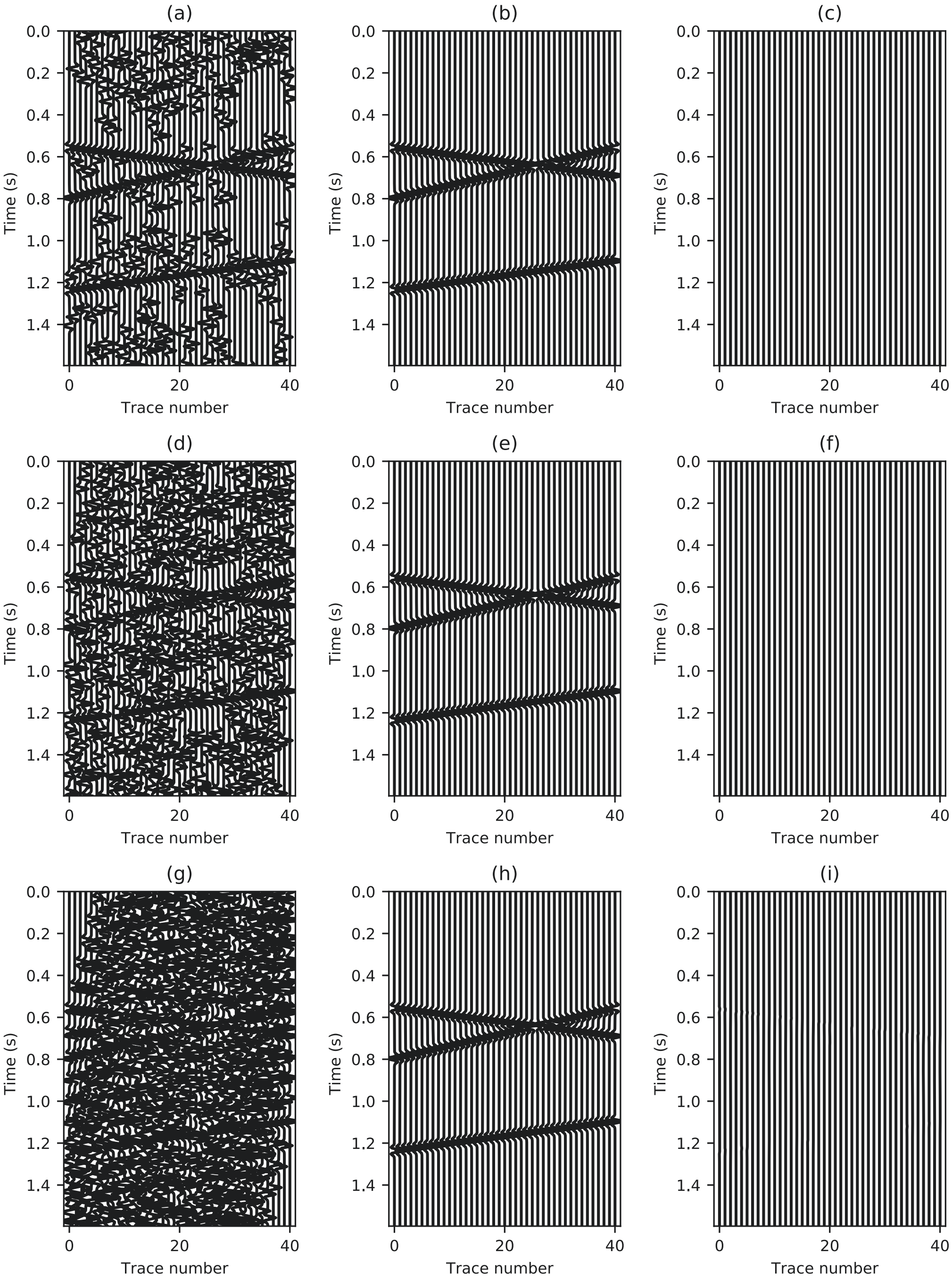

Figure 2.9: Denoising results for the ℓ_p estimator with $p = 1$. (a) Noisy data with a blending factor of 5. (b) Denoised data $SNR = 123.4dB$. (c) Errors between b and the clean data. (d) Noisy data with a blending factor of 10. (e) Denoised data $SNR = 113.8$ dB. (f) Errors between e and the clean data. (g) Noisy data with a blending factor of 20. (h) Denoised data $SNR = 34.8$ dB. (i) Errors between h and the clean data.

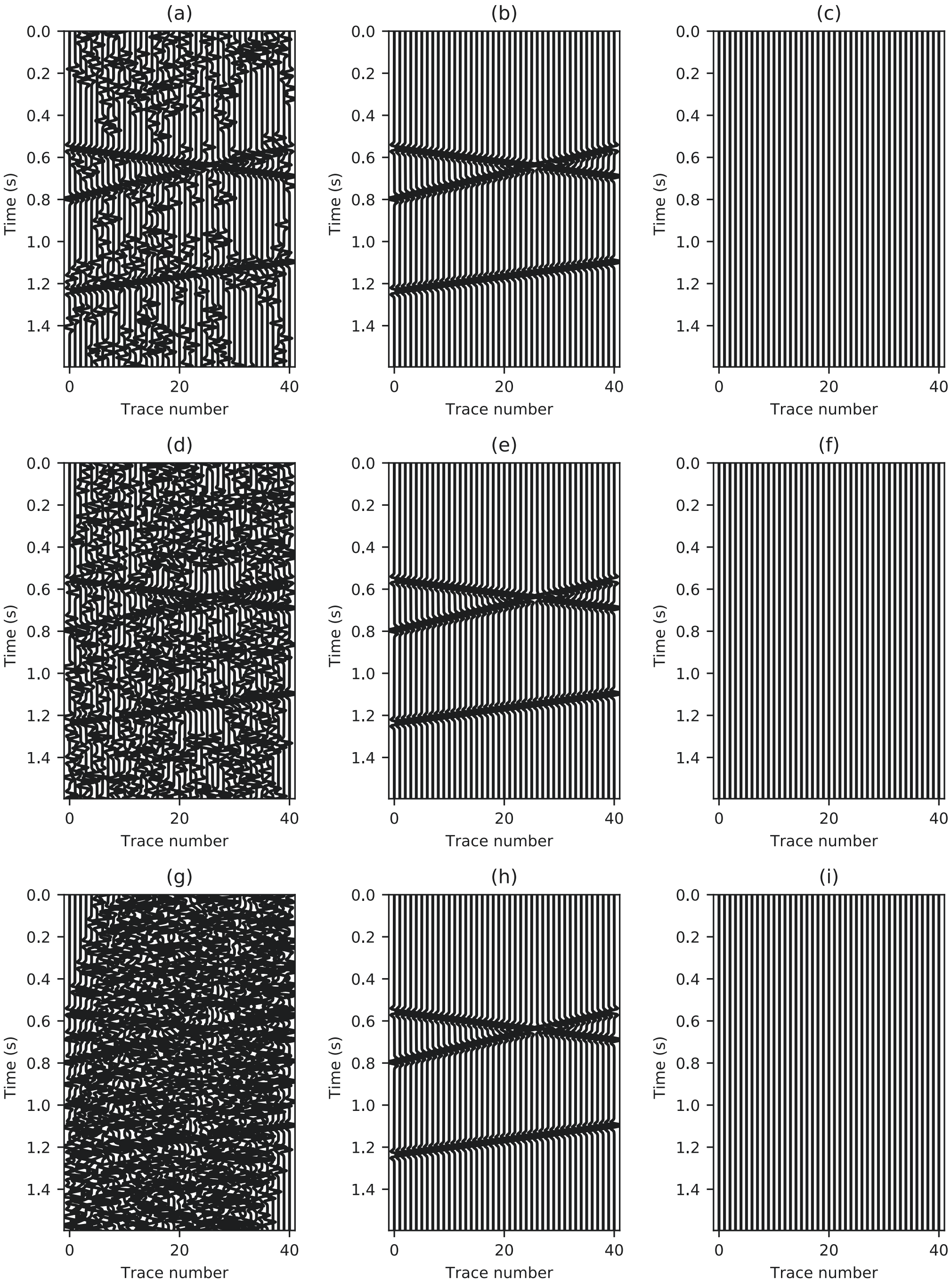

Figure 2.10: Denoising results for the ℓ_p estimator with $p = 0.8$. (a) Noisy data with a blending factor of 5. (b) Denoised data $SNR = 142.2$dB. (c) Errors between b and the clean data. (d) Noisy data with a blending factor of 10. (e) Denoised data $SNR = 138.8$ dB. (f) Errors between e and the clean data. (g) Noisy data with a blending factor of 20. (h) Denoised data $SNR = 120.9$ dB. (i) Errors between h and the clean data.

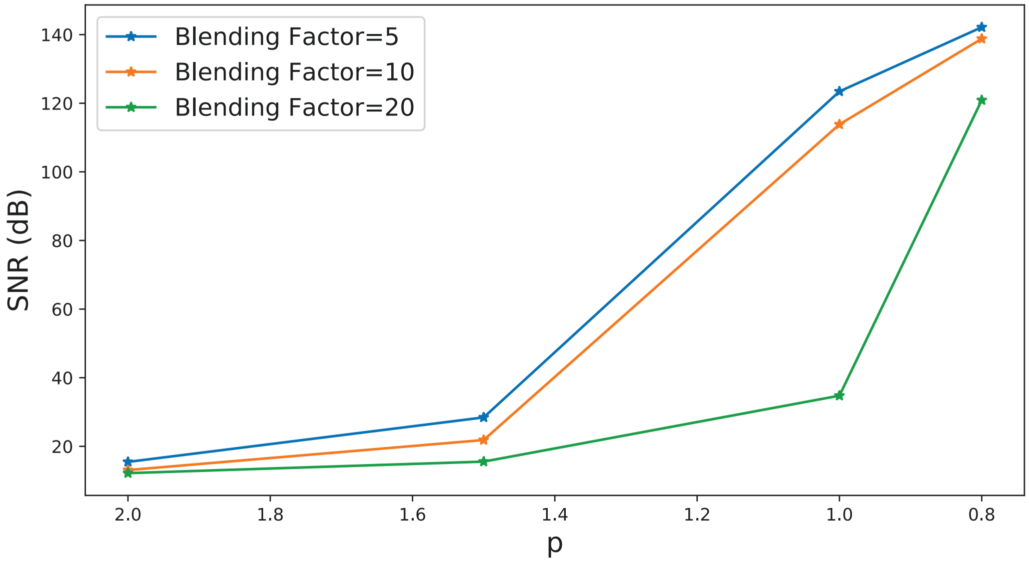

Figure 2.11: Comparison of denoising performance versus the p-norm parameter p.

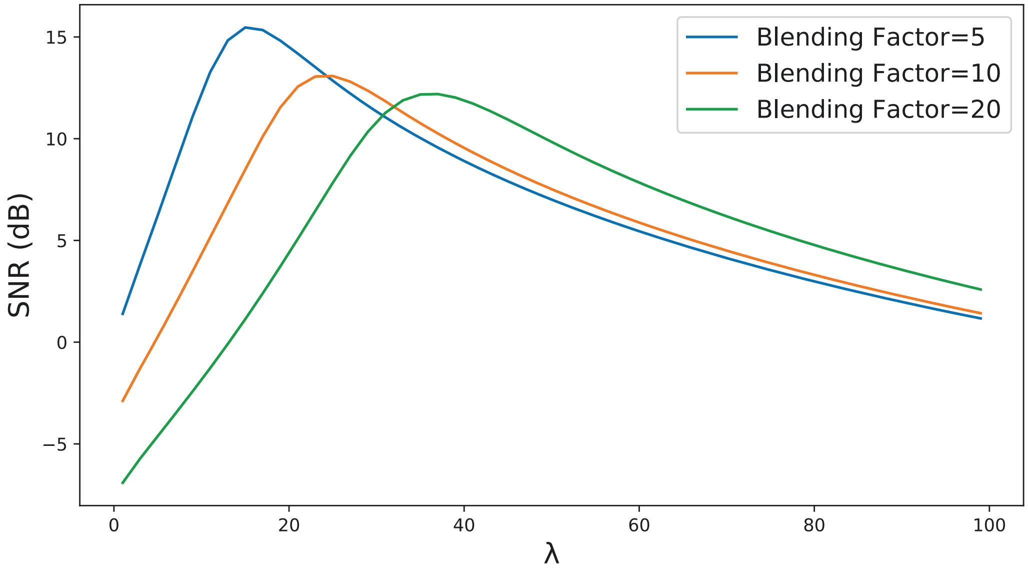

Figure 2.12: Denoising performance vs λ for p-norm parameter $p = 2$.

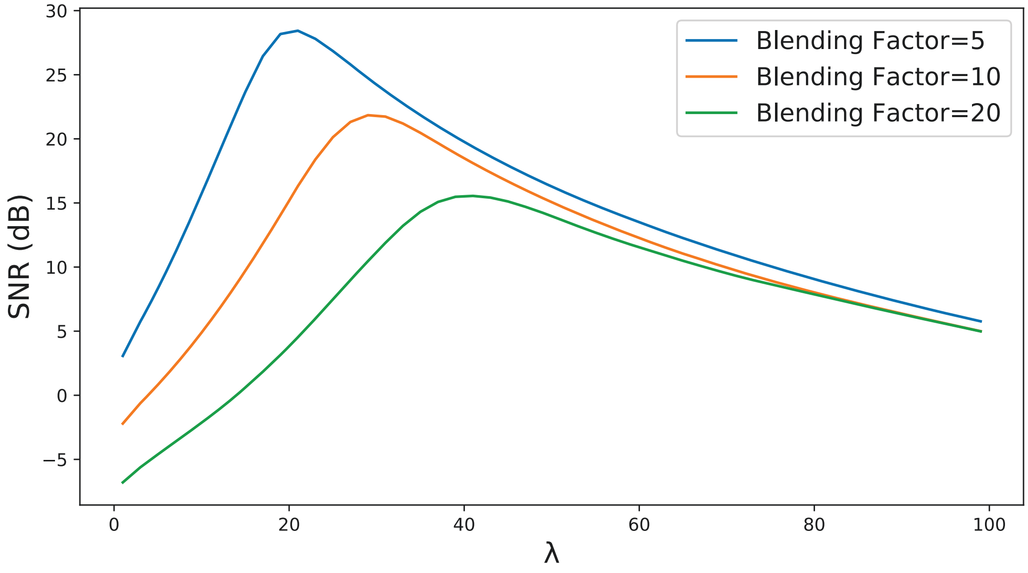

Figure 2.13: Denoising performance vs λ for p-norm parameter $p = 1.5$.

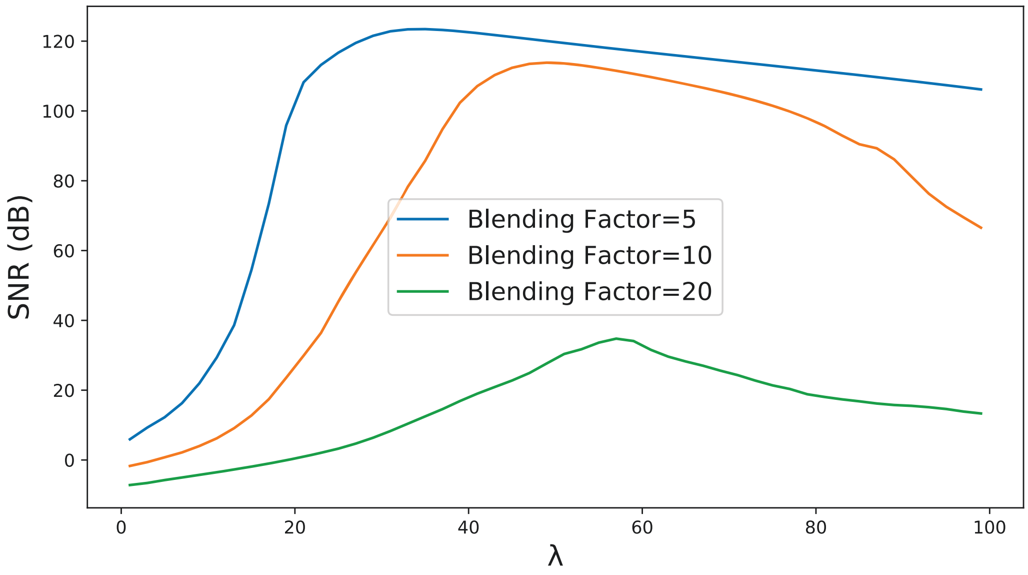

Figure 2.14: Denoising performance vs λ for p-norm parameter $p = 1$.

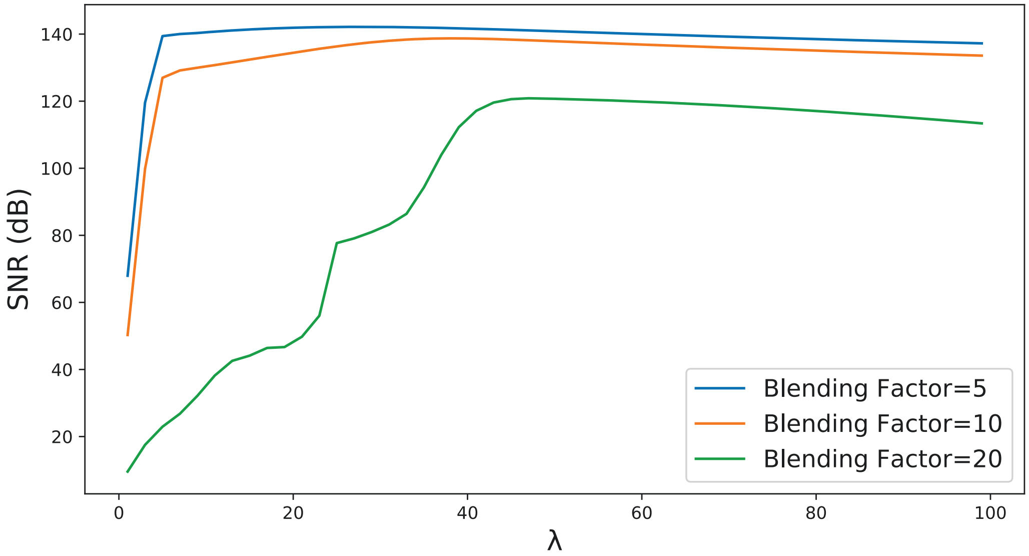

Figure 2.15: Denoising performance vs λ, for p-norm parameter $p = 0.8$.

CHAPTER 3

Deblending via robust matching pursuit [1]

In chapter 1, I introduced compressive sensing and sparse reconstruction. Then, I discussed different robust sparse reconstruction methods. In this chapter, I propose a new robust and sparse estimation technique based a robust Matching Pursuit algorithm. I will also show how to use the robust Matching pursuit to solve the simultaneous source separation problem via Radon transforms. First, we need to explain the theory of Matching pursuit in detail.

3.1 Theory

3.1.1 Matching pursuit

The Matching Pursuit method iteratively represents a signal via a superposition of simple components. After k iterations, the signal is described by the following

[1] This chapter contains material published in J Li and M D Sacchi, 2021, An lp-space Matching Pursuit algorithm and its application to robust seismic data denoising via time-domain Radon transforms: Geophysics, in Press.

expressions

$$\mathbf{d} = \mathbf{s}^{[k]} + \mathbf{r}^{[k]}, \tag{3.1}$$

$$\mathbf{s}^{[k]} = \sum_{j \in T^{[k]}} a_j \, \mathbf{g}_j \, , \tag{3.2}$$

where $\mathbf{s}^{[k]}$ is the reconstructed or synthesized signal and $\mathbf{r}^{[k]}$ is the reconstruction error. In the influential paper introducing Matching Pursuit (Mallat and Zhang, 1993), the algorithm is summarized by the following process: starting with an initial approximation $\mathbf{s}^{[0]} = \mathbf{0}$, an empty set $T^{[0]} = \{\}$ and residual $\mathbf{r}^{[0]} = \mathbf{d}$, the subsequent iteration is used to update $\mathbf{r}^{[k]}$ and $\mathbf{s}^{[k]}$

$$l = \underset{j=1,2,\ldots,M}{\operatorname{argmax}} \; |\mathbf{g}_j^H \mathbf{r}^{[k]} \; |, \tag{3.3}$$

$$a_l = \underset{a}{\operatorname{argmin}} \; \|\mathbf{r}^{[k]} - a \, \mathbf{g}_l \, \|_2^2, \tag{3.4}$$

$$T^{[k+1]} = T^{[k]} \cup \{l\}, \tag{3.5}$$

$$\mathbf{r}^{[k+1]} = \mathbf{r}^{[k]} - a_l \, \mathbf{g}_l, \tag{3.6}$$

$$\mathbf{s}^{[k+1]} = \mathbf{s}^{[k]} + a_l \, \mathbf{g}_l \, . \tag{3.7}$$

In each iteration, the algorithm selects the basis function that maximizes the absolute value of the inner product between the residual $\mathbf{r}^{[k]}$ and all basis functions that belong to the dictionary D (Equation 3.3). Then, once the best basis function is selected, the optimal coefficient a_l is obtained via the method of least-squares (equation 3.4). Notice that equation 3.4 has a simple closed-form solution given by

$$a_k = \frac{\mathbf{g}_l^H \mathbf{r}^{[k]}}{\|\mathbf{g}_l\|_2^2} \, . \tag{3.8}$$

The expression given by equation 3.8 is a normalized l_2 inner product between the vector $\mathbf{r}^{[k]}$ and the vector $\mathbf{g}_l$. For a normalized dictionary with elements satisfying $\|\mathbf{g}_l\|_2^2 = 1$, the coefficient becomes $a_l = \mathbf{g}_l^H \mathbf{r}^{[k]}$.

3.1.2 Robust Matching pursuit

The l_2 inner product in the classical Matching Pursuit algorithm (equation 3.8) is sensitive to data outliers. Many authors have observed that when the observations $\mathbf{d}$ are contaminated by erratic noise, the l_2 inner product could result in the selection of an incorrect basis function (Razavi et al., 2012; Chen et al., 2013; Moore et al., 2016). The problem can be alleviated by modifying the steps given by equations 3.3 and 3.4 of the classical Matching Pursuit algorithm. One can interpret the l_2 inner product between the vector $\mathbf{r}$ and a waveform $\mathbf{g}$ (notice that we have removed the sub-index l from $\mathbf{g}_l$ to avoid notational clutter) as the solution of the following problem

$$\begin{aligned} (\mathbf{g}, \mathbf{r})_2 := a_{opt} &= \underset{a}{\operatorname{argmin}} \; \|\mathbf{r} - a\,\mathbf{g}\,\|_2^2 \\ &= \frac{\mathbf{g}^H \mathbf{r}}{\mathbf{g}^H \mathbf{g}} \,. \end{aligned} \tag{3.9}$$

Inspired by the last equation, we propose to estimate the following l_p inner product via the following expression

$$\begin{aligned} (\mathbf{g}, \mathbf{r})_p := a_{opt} &= \underset{a}{\operatorname{argmin}} \; \|\mathbf{r} - a\,\mathbf{g}\,\|_p^p \\ &= \frac{\mathbf{g}^H \mathbf{W} \mathbf{r}}{\mathbf{g}^H \mathbf{W} \mathbf{g}} \,, \end{aligned} \tag{3.10}$$

where $\mathbf{W}$ is a diagonal matrix with elements given $w_i = (|d_i - a g_i|^{2-p} + \epsilon)^{-1}$. Equation 3.10 does not have a closed-form solution for $p < 2$. In this case, we adopt the Iteratively Reweighed Least-Squares (IRLS) method (Scales and Gersztenkorn, 1988) to compute $(\mathbf{g}, \mathbf{r})_p$ (see Appendix). The small number $\epsilon > 0$ in w_i avoids division by zero. We name the proposed method for $p \approx 1$ the Robust Matching Pursuit algorithm. The pseudo-code is given by Algorithm 1. Notice that the algorithm becomes the classical Matching Pursuit method for $p = 2$. At this stage, it is important to mention that the IRLS algorithm adopted to evaluate the l_p inner product convergences in about $4 - 5$ iterations. Also notice that $(\mathbf{g}, \mathbf{r})_p$ is not an inner product in a strict mathematical definition (Zeng et al., 2016).

We now illustrate the problem via a simple example which is provided in Figure 3.1. The signal corresponds to a complex exponential given by $d(n) = e^{i\,(n-1)\,\tilde{k}}, n = 1 \ldots N$ with non-dimensional wavenumber $\tilde{k} = 0.2$ radians and length $N = 100$ samples. The clean signal is provided in Figure 3.1(a). The signal contaminated by three outliers is portrayed in Figure 3.1(b). In this example, the dictionary D is given by complex exponentials $g_j(n) = e^{i(n-1)\,k_j}$ of non-dimensional wavenumber $k_j = -\pi + 2\pi(j-1)/(M-1)$ where $n = 1 \ldots N$, $j = 1 \ldots M$ and $M = 250$. We run the classical Matching Pursuit algorithm for $K = 20$ iterations, and we displayed the recovered data in Figure 3.1(c). The absolute value of the retrieved Fourier coefficients, $|a_k|$, are plotted versus wavenumber in the interval $[-\pi, \pi]$ in Figure 3.1(d). The classical Matching Pursuit algorithm identifies the correct basis function and includes spurious spectral peaks required to fit the outliers. The recovered data show the persistence of outliers in the solution. Additionally, Figure 3.1(e) and Figure 3.1(f) show the reconstructed data and the identified Fourier coefficients for the robust Matching Pursuit algorithm with $p = 0.8$, respectively. In this case, the algorithm is also run for $K = 20$ iterations. The algorithm now selects the correct basis function in the first iteration. In the remaining iterations, the robust selection of the basis function leads to $a_k = 0$. In other words, as indicated by our analysis of the convergence diagram in Figure 3.2, the classical Matching Pursuit method adds spurious harmonics to further decrease the norm of the residuals. Conversely, immediately after one iteration, the robust Matching Pursuit algorithm ceases to add new harmonics. In essence, the absence of spurious harmonics (Figure 3.1(e) and (f)) and the flattening of the convergence curve (Figure 3.2) indicate that the robust Matching Pursuit algorithm can tolerate outliers.

Algorithm 1 Matching Pursuit with l_p inner product

```
Input: data $\mathbf{d}$, dictionary $D$ with normalized atoms $\mathbf{g}_j, j = 1 \ldots M$
Output: reconstructed data $\mathbf{d}_r^{[K]}$ and residuals $\mathbf{r}^{[K]}$
Initialization: $\mathbf{r}^{[0]} = \mathbf{d}$, $\mathbf{d}_r^{[0]} = 0$ and $T^{[0]} = \{\}$
Select $p = 2$ for classical MP or $p \approx 1$ for robust MP
for $k = 1, 2, ...., K$ do
    $l = \underset{j=1,2,....M}{\text{argmax}} \; |(\mathbf{r}^{[k-1]}, \mathbf{g}_j)_p|$
    $\alpha = (\mathbf{r}^{[k-1]}, \mathbf{g}_l)_p$
    $T^{[k]} = T^{[k-1]} \cup \{l\}$
    $\mathbf{r}^{[\mathbf{k}]} = \mathbf{r}^{[k-1]} - \alpha\, \mathbf{g}_l$
    $\mathbf{d}_r^{[k]} = \mathbf{d}_r^{[k-1]} + \alpha\, \mathbf{g}_l$
end for
```

3.1.3 Computing sparse Radon transforms via Robust Matching Pursuit

We now apply the proposed robust Matching Pursuit algorithm to the computation of the Radon transform. Unfortunately, the time domain Radon transform entails solving an inverse problem in an implicit formulation (Trad et al., 2003b). Simply speaking, one does not have access to explicit Radon basis functions in a matrix-vector form. Instead, we need to apply the robust inner product inside the Radon operator, I already discribed the Radon operator in the Chapter 2. I will shows the

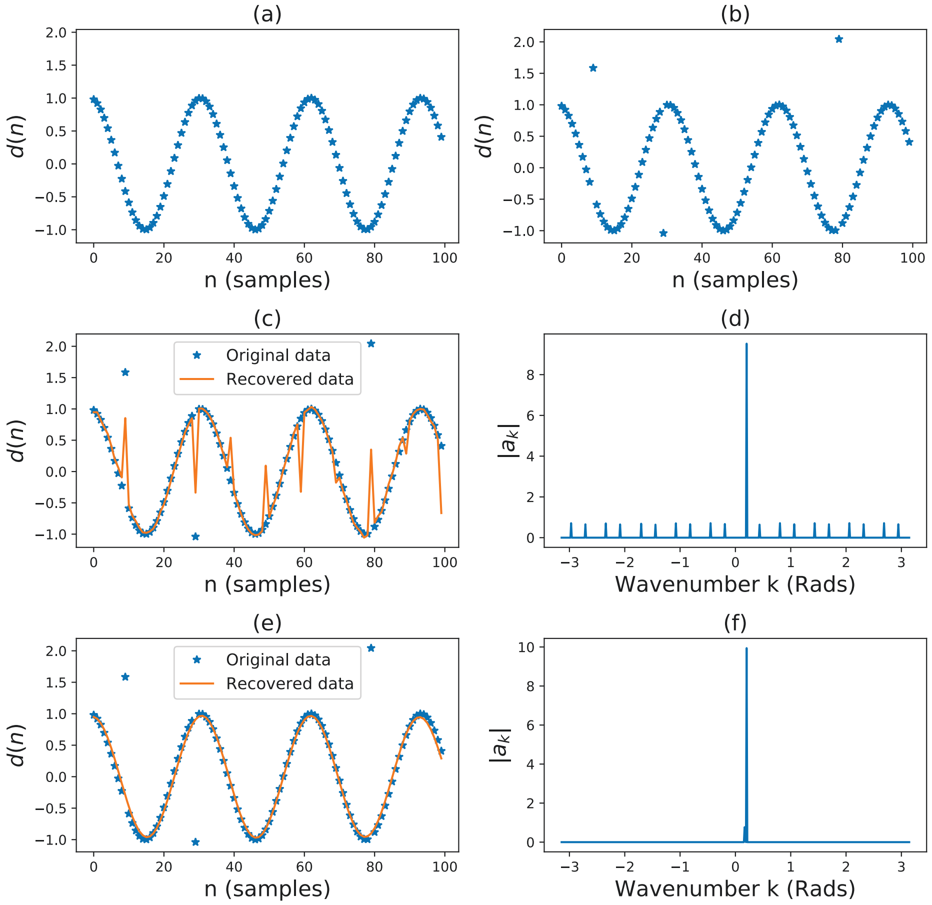

Figure 3.1: One-dimensional Fourier example. (a) The original data consists of one harmonic of non-dimensional wavenumber 0.2 rads. (b) The original data plus outliers. (c) The estimated data using the classical Matching Pursuit algorithm with l_2 inner products. Blue stars indicate the original data and the orange solid line for recovered data. (d) The Fourier coefficients (a_k) that were identified by the classical non-robust Matching Pursuit algorithm. (c) Estimated data using the proposed robust Matching Pursuit algorithm with l_p inner product ($p = 0.8$). The blue stars represent the original data, and the orange solid line represents the recovered data. (d) The absolute value of the Fourier coefficients a_k that were identified by the Robust Matching Pursuit algorithm.

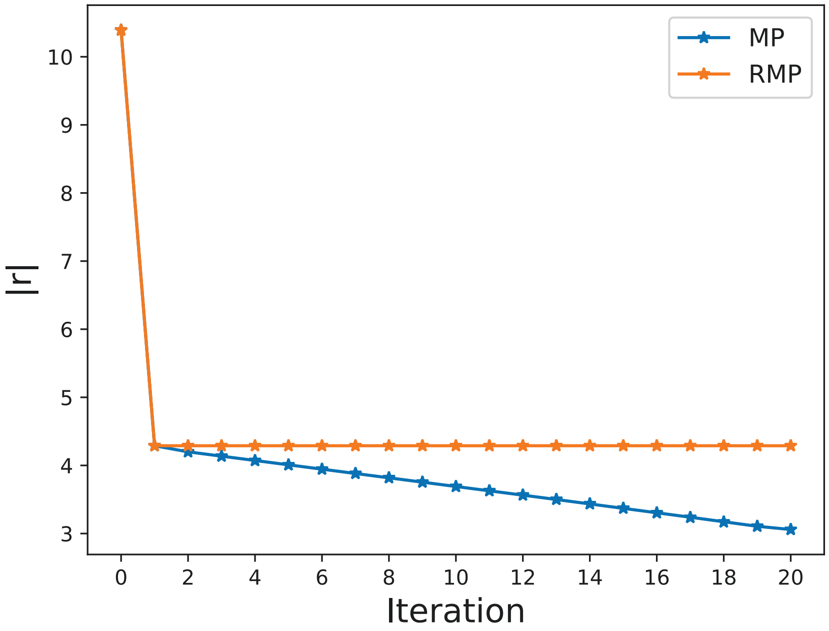

Figure 3.2: Convergence curves for the example provided in Figure 1. The vertical axis is the norm of the residual vector. The figure compares the convergence curves of the classical Matching Pursuit (MP) algorithm versus the Robust Matching Pursuit (RMP) algorithm proposed in this paper. The blue line indicates the convergence curve for the standard Matching Pursuit algorithm. The orange line shows the convergence curve for the Robust Matching Pursuit algorithm.

equations for the Radon operators here again.

$$L : d(t,h) = \sum_q a(\tau = t - q\,\phi(h), q), \tag{3.11}$$

$$L^* : \tilde{a}(\tau,q) = \sum_h d(t = \tau + \phi(h)q, h), \tag{3.12}$$

We notice that the adjoint Radon operator L^* can be interpreted as the inner product between data extracted across each τ, q trajectory and an all ones-vector $\mathbf{1}$. We denote the data extracted along a τ, q trajectory as the length $n_h \times 1$ vector given by

$$\mathbf{u}(\tau,q) = [d(\tau + q\,\phi(h_1), h_1), d(\tau + q\,\phi(h_2), h_2), \ldots, d(\tau + q\,\phi(h_{n_h}), h_{n_h})]^T \,. \tag{3.13}$$

Hence, we can write

$$\tilde{a}(\tau,q) = (\mathbf{u}(\tau,q), \mathbf{1})_p \,. \tag{3.14}$$

For $p \approx 1$, the Matching Pursuit algorithm will become insensitive to outliers and, therefore, less prone to the selection of the incorrect basis function. Algorithm 2 provides the proposed robust Matching Pursuit algorithm for the Radon transform. The algorithm follows the same logic of Algorithm 1 but now one needs to be more creative in the definition of its parts because we do not have access to Radon transform basis functions in explicit form. Notice that when $p = 2$, Algorithm 2 is the classical Matching Pursuit algorithm which is non-robust and can be adopted to estimate sparse Radon coefficients. Also, notice that in Algorithm 2, the Radon coefficients with maximum absolute l_p inner product is used to model one individual basis function via the forward Radon operator. We also mention that the symbol $\delta(t,q)$ in Algorithm 2 is the Kronecker delta function which serves to model one individual waveform. The modelled resulting waveform is then fitted to the data.

Figure 2 indicates that, for the robust Matching Pursuit algorithm, the norm of the residuals does not change after all the correct basis functions were selected. For this reason, we have adopted the following stopping criteria. The algorithm stops either when it reaches a maximum number of iterations or the norm of the relative change of the residuals satisfies $\|\mathbf{r}^k - \mathbf{r}^{k-1}\|/|\mathbf{r}^{k-1}\| < \eta$ where $\eta = 0.05$. These two criteria were used for all numerical experiments run with classical and robust Matching Pursuit methods.

Algorithm 2 Robust Matching Pursuit Radon transform

```
Input: data $\mathbf{d}$, forward $L$ and adjoint $L^*$ Radon operators
Output: reconstructed data $\mathbf{d}_r^{[K]}$ and residuals $\mathbf{r}^{[K]}$
Initialization: $\mathbf{r}^{[\mathbf{0}]} = \mathbf{d}$, $\mathbf{d}_r^{[0]} = 0$ and $T^{[0]} = \{\}$
for $k = 1, 2, ...., K$ do
    for all $\tau, q$ do
        $\mathbf{u}(\tau, q) = [d(\tau + q\,\phi(h_1), h_1), d(\tau + q\,\phi(h_2), h_2), \dots, d(\tau + q\,\phi(h_{n_h}), h_{n_h})]^T$
        $\tilde{a}(\tau, q) = (\mathbf{u}(\tau, q), \mathbf{1})_p$
    end for
    $(\tau_l, q_l) = \underset{\tau,q}{\operatorname{argmax}} |\tilde{a}(\tau, q)|$
    $a(\tau, q) = \delta(\tau - \tau_l, q - q_l)$
    $\mathbf{g} = L\,a(\tau, q)$
    $\alpha = \underset{\alpha'}{\operatorname{argmin}} \|\mathbf{r}^{[k-1]} - \alpha' \mathbf{g}\|_p^p$
    $T^{[k]} = T^{[k-1]} \cup \{l\}$
    $\mathbf{r}^{[\mathbf{k}]} = \mathbf{r}^{[k-1]} - \alpha \mathbf{g}$
    $\mathbf{d}_r^{[k]} = \mathbf{d}_r^{[k-1]} + \alpha \mathbf{g}$
    $k \leftarrow k + 1$
end for
```

3.2 Examples

3.2.1 First example

Using the parabolic forward Radon operator, we synthesize data that consists of eight parabolic events. The spatial sampling is 51 m, and the temporal sampling interval is 0.004 s. We also use a zero-phase Ricker wavelet of central frequency $f_c =$ 20 Hz to simulate the source wavelet. Figure 3.3a shows the clean data. Figure 3.3b is the data after contamination with erratic noise. The noise imitates the type of data corruption that one will encounter in a simultaneous source acquisition environment. The gather can be considered a pseudo-deblended common receiver gather where the noise is erratic. We define the input signal-to-noise ratio as $SNR_{in} = 10 \log \frac{\|\mathbf{d}_c\|_2^2}{\|\mathbf{n}_e\|}$ where $\mathbf{d}_c$ is the clean signal and $\mathbf{n}_e$ is the erratic noise. For this example $SNR_{in} =$ -8.1 dB.

We test the retrieval of Radon coefficients and data reconstruction by first adopting the classical non-robust Matching Pursuit algorithm ($p = 2$). Then we investigate the proposed robust Robust Matching Pursuit method with parameter $p = 0.8$. In this example, we use the parabolic Radon operator. We run the algorithm for a maximum of $K = 20$ iterations.

In Figure 3.3c and Figure 3.3d we show the reconstructed data and the reconstruction error for the classical Matching Pursuit algorithm ($p = 2$). Similarly, Figure 3.3e and Figure 3.3f show the reconstructed data and reconstruction error obtained via the robust Matching Pursuit algorithm with parameter $p = 0.8$. We also computed the signal-to-noise-ratio of the output $SNR_{out} = 10 \log \frac{\|\mathbf{d}_c\|_2^2}{\|\mathbf{d_c} - \mathbf{d_r}\|}$ where $\mathbf{d}_r$ is the reconstructed data. For this example $SNR_{out} = 77$ dB. We also save the sparse Radon coefficients $a(\tau, q)$ identified by the Matching Pursuit algorithm. Figure 3.4a shows the true parabolic Radon coefficients adopted to synthesize the data in Figure 3.3a. Figure 3.4b shows the coefficients identified by the classical non-robust Matching Pursuit algorithm. Finally, Figure 3.4c shows the coefficients identified by the robust Matching Pursuit algorithm with $p = 0.8$. We also need to mention that the retrieved sparse coefficients were convolved with a two-dimensional smoothing kernel to facilitate their visualization. We find that the robust Matching Pursuit algorithm avoids picking the wrong basis functions caused by erratic noise.

3.2.2 Second example

We also compare our proposed algorithm with a different deblending method: iterative rank reduction (IRR) deblending first studied by Cheng and Sacchi (2015). The rank reduction filter of IRR is based on the Singular Spectrum Analysis (SSA) method, which is iteratively implemented in local windows as a denoiser. The IRR deblending method falls in the deblending by inversion category, where recorded simultaneous source data are inverted to estimate the deblended data (Abma et al., 2010; Chen et al., 2014; Abma et al., 2015). The proposed robust Matching Pursuit method belongs to the category of deblending by denoising methods (Huo et al., 2012; Ibrahim and Sacchi, 2014), which in general, are easier to implement and can be relatively faster than deblending via inversion techniques. Figure 3.5 shows a comparison between the IRR deblending method and robust Matching Pursuit with the linear Radon transform operator. We display three windows corresponding to three linear events of different dips (Figure 3.5a, e and i). The first window, Figure 3.5a, contains three distinct dips. The second window, Figure 3.5e, contains three dips, and two dips are similar. In the third window, Figure 3.5i, two events are spatially aliased. We adopted a rank $P = 3$ for IRR. For the proposed robust Matching Pursuit method, we use $p = 0.8$ and the maximum number of iterations $K = 20$ and

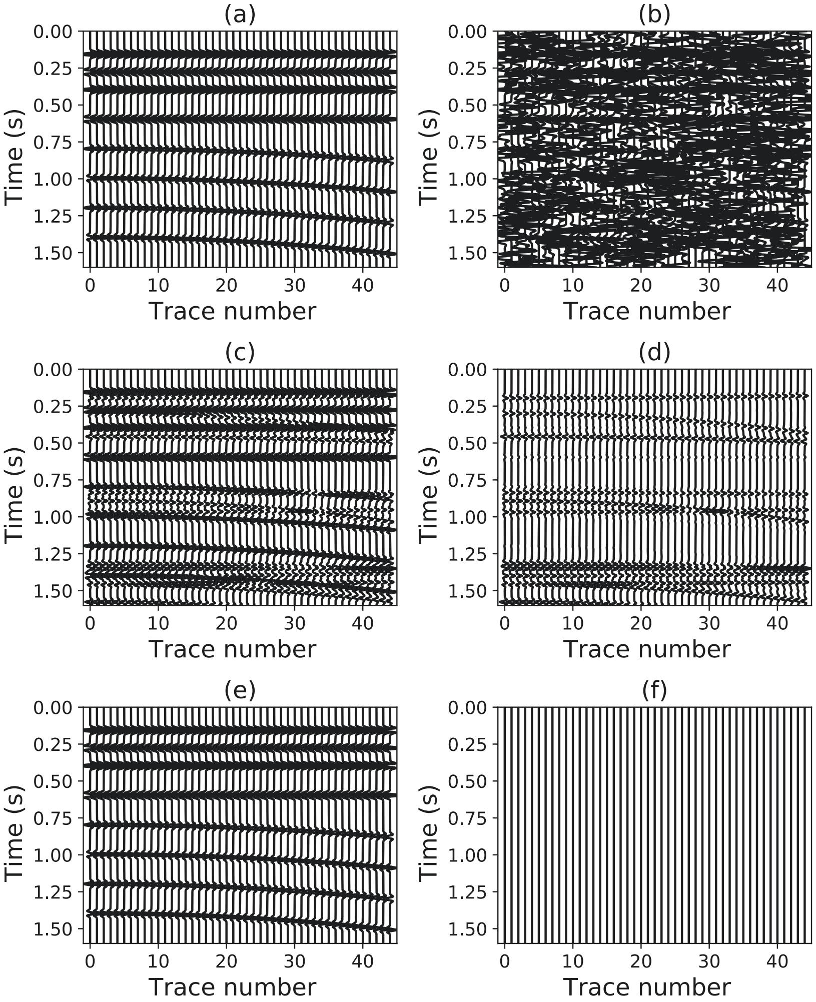

Figure 3.3: (a) Clean data synthesized via a foward parabolic Radon transform. (b) Data contaminated with erratic noise, $SNR_{in} = -8.1$ dB. (c) Reconstruction of the data via classical (non-robust) Matching Pursuit, $SNR_{out} = 8.0$ dB. (d) Error panel given by reconstructed data in (c) minus the clean data. (e) Reconstruction via the proposed robust Matching Pursuit algorithm, $SNR_{out} = 77$ dB. (f) Error panel given by the reconstructed data in (e) minus the clean data.

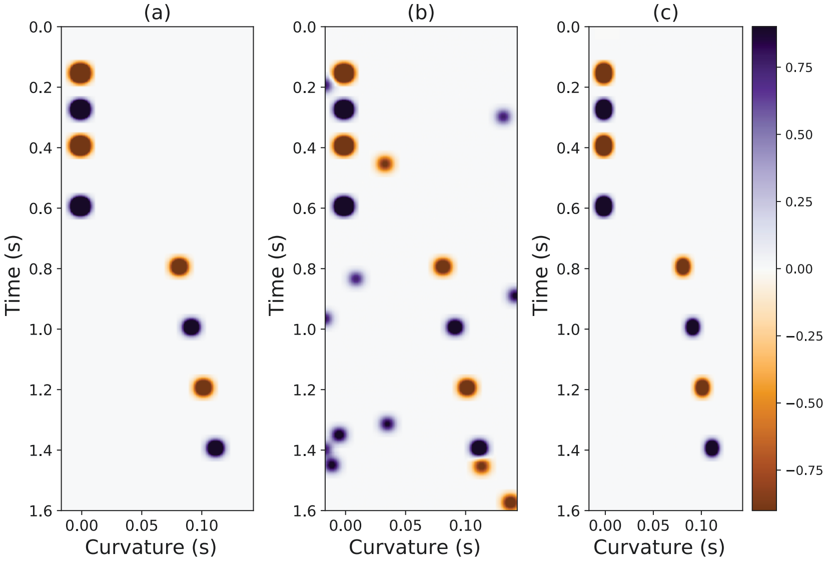

Figure 3.4: (a) Parabolic Radon coefficients utilized to synthesize the data in Figure 3.3a. (b) Coefficients retrieved via the classical (non-robust) Matching Pursuit method. (c) Coefficients retrieved via the proposed robust Matching Pursuit algorithm.

tolerance $\eta = 0.05$. Figure 3.5b,f and j show the data contaminated with erratic blending noise. Figure 3.5c, g and k show the resulting deblended data via IRR. Figure 3.5d, h and l correspond to the data deblended via robust Matching Pursuit. We observe that the proposed method produces results with fewer artifacts than IRR. For completeness, we also compare IRR and the proposed robust Matching Pursuit Radon denoising method in Figure 3.6. In this case, we evaluate SNR_{out} versus blending factor for randomly generated events of varying dips similar to those portrayed in Figure 3.6a,e and i. As the blending factor increases, we expect the quality of the construction to decrease. However, we observe that the quality of the reconstruction measured by SNR_{out} does not deteriorate too dramatically for the robust Matching Pursuit Radon denoising method.

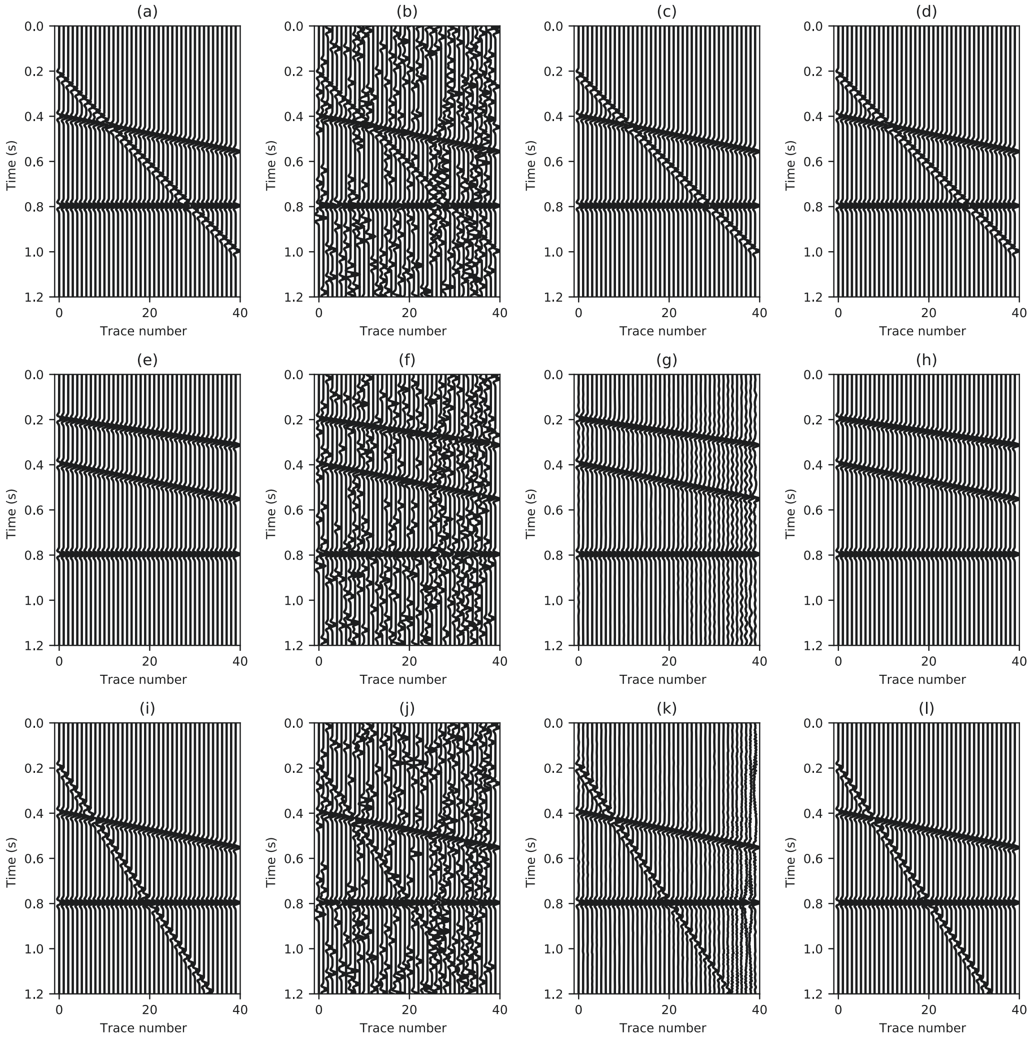

Figure 3.5: (a) Clean window. (b) Pseudo-deblended data. (c) Deblending via iterative rank reduction (IRR) with resulting $SNR = 27.4$. d) Deblending via robust Matching Pursuit ($SNR = 32.$). (e) Clean window. (f) Pseudo-deblended data. (g) Deblending via IRR ($SNR = 5.3$). h) Deblending via robust Matching Pursuit ($SNR = 17.2$) (i) Clean window. (j) Pseudo-deblended data. (k) Deblending via IRR ($SNR = 4.7$). l) Deblending via robust Matching Pursuit ($SNR = 14.2$).

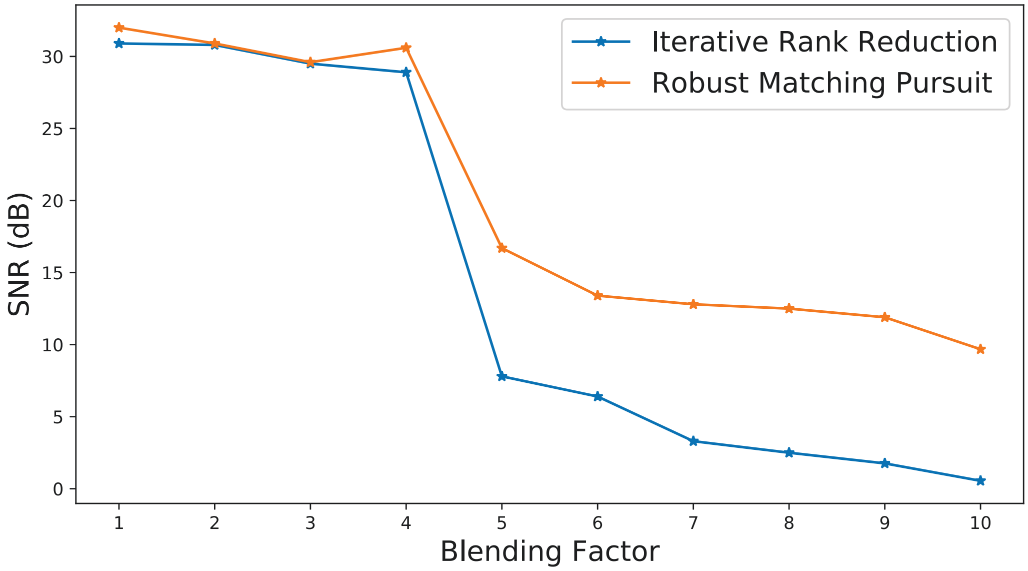

Figure 3.6: The blue line represents the deblending via iterative rank reduction (IRR) and the orange line shows deblending via the proposed robust Matching Pursuit method.

3.2.3 Acoustic finite differences synthetic example

We also tested the proposed method with a 2D synthetic data example generated by an acoustic finite-difference method. The velocity model used to create this example is shown in Figure 3.7. The total length of the model is 7000 m, and the depth is 3000 m. A total of 350 receivers are evenly distributed at a 500 m depth below the sea level to simulate an ocean bottom survey. Moreover, 350 sources are triggered at the surface of the ocean to simulate the survey. Acoustic finite differences data were numerically blended to mimic a simultaneous source survey. For this purpose, we set five adjacent sources as a group and fire these sources with a short time difference ranging between 0 and 2 s. A total number of 70 source groups were simulated. By this type of acquisition, the survey time is about one-fifth of a conventional survey acquisition time with the same number of sources.

The acquisition geometry, including firing times, is provided in Figure 3.8.

Since the linear events assumption only valid for a small patch of seismic data, we can define a window operator that combines the linear radon transform to excise the local radon transform. The forward window operator decomposes the common receiver gathers into small overlapping windows, and the adjoint window operator will combine all small patches back into the common receiver gathers. And a Gaussian taper is used in the overlapping area between the windows. For both finite difference and real data example, we set the window size equal to 10 traces and 400 time samples per window, and apply 20% overlap along each edge. Unlike the simple synthetic example, both finite differences and real data have thousands of non-zero coefficients needing to be reconstructed. The main computational cost for this Matching Pursuit algorithm is coming from each time we calculate the adjoint Radon coefficients. Therefore, to reduce the total time costs, we apply a simple trick here; instead of picking 1 radon coefficients in each iteration, we select the largest N coefficients. Then we fit these coefficients via the method like conjugate gradient for one-time. We apply this method to both finite difference examples and real marine data example.

Figure 3.9a shows an ideal common receiver gather that we would have recorded via a conventional acquisition. Figure 3.9b shows the common receiver gather for the simultaneous source survey. This is often referred to as the pseudo-deblended data. We adopted a local linear Radon transform we explained above to remove the blending noise observable in Figure 3.9b. Figure 3.9c and Figure 3.9d show the reconstructed data and reconstruction error for the non-robust Radon transform with coefficients extracted via classical Matching Pursuit. These results are not very encouraging. The non-robust Marching pursuit algorithm was unable to reduce the blending noise from the data. Conversely, Figure 3.9e and 3.9f show an excellent

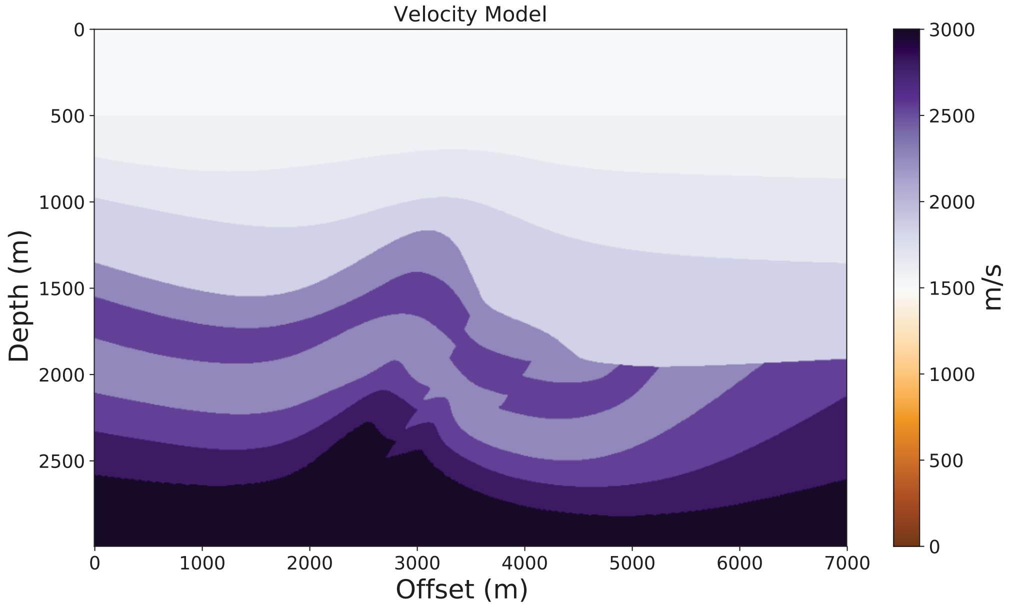

Figure 3.7: The velocity model used to generate synthetic data via acoustic finite differences.

recovery of the data with a significant attenuation of blending noise.

We conducted erratic noise attenuation to all common receiver gathers, and when the process was finalized, the data was sorted back into common shot gathers. Figure 3.10 shows the results for one common shot gather. For this particular example, the $SNR_{in} = -3.6$ dB. The resulting deblended data via classical Matching Pursuit yield $SNR_{out} = -3.4$ dB, which indicates that the non-robust Matching Pursuit failed at removing blending noise. Denoising via the robust Matching Pursuit algorithm (Figures 3.9e and 3.10e) has lead to a result with $SNR_{out} = 18.8$ db.

3.2.4 Field data example

Finally, we test our algorithm on the real marine dataset that we numerically blended to simulate a simultaneous source survey. This is a subset of a marine seismic data

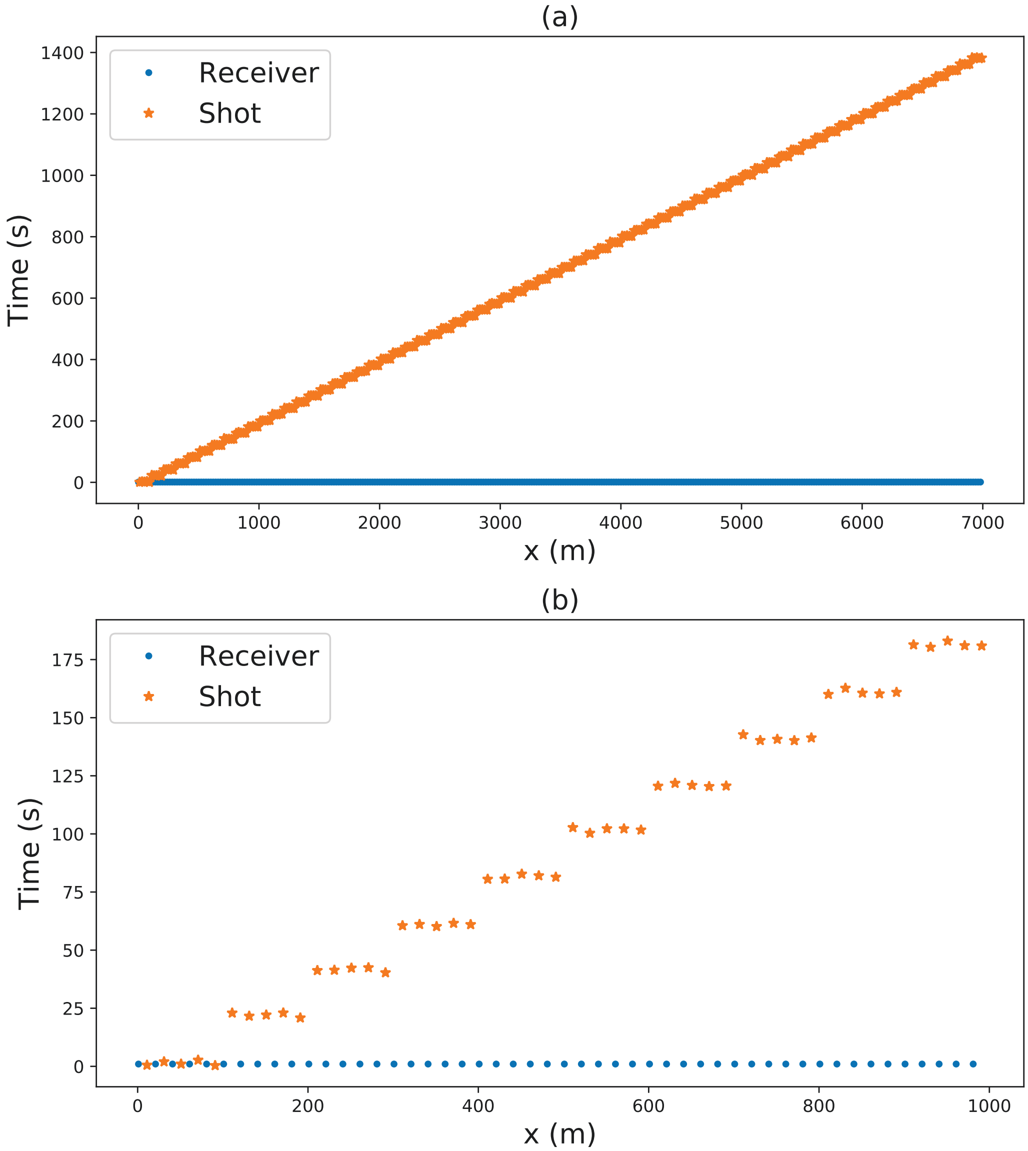

Figure 3.8: (a) Distribution of receivers (dots) and sources (stars). Note that the vertical axis indicates source firing time, . (b) Detailed window of (a). The acquisition simulates a two-dimensional ocean bottom survey with 70 group of sources of 5 sources each.

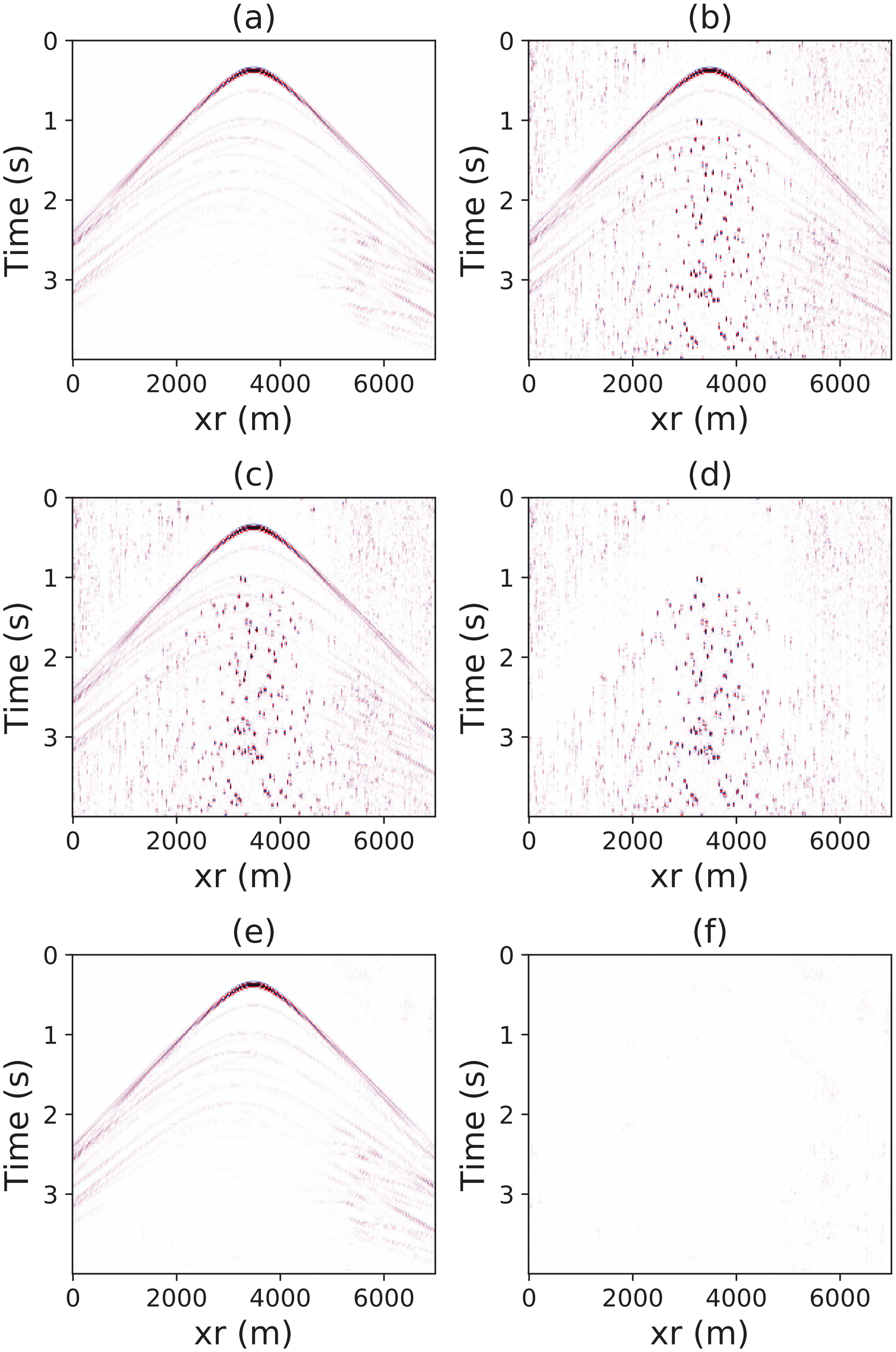

Figure 3.9: (a) One ideal common receiver gather. (b) Pseudo-deblended data common receiver gather $SNR_{in} = -1.73$ dB. (c) Deblending via classical Matching Pursuit $SNR_{out} = -1.40$ dB. (d) Difference between (a) and (c). (e) Deblending via robust Matching Pursuit $SNR_{out} = 17.8$ dB. (f) Difference between (a) and (e).

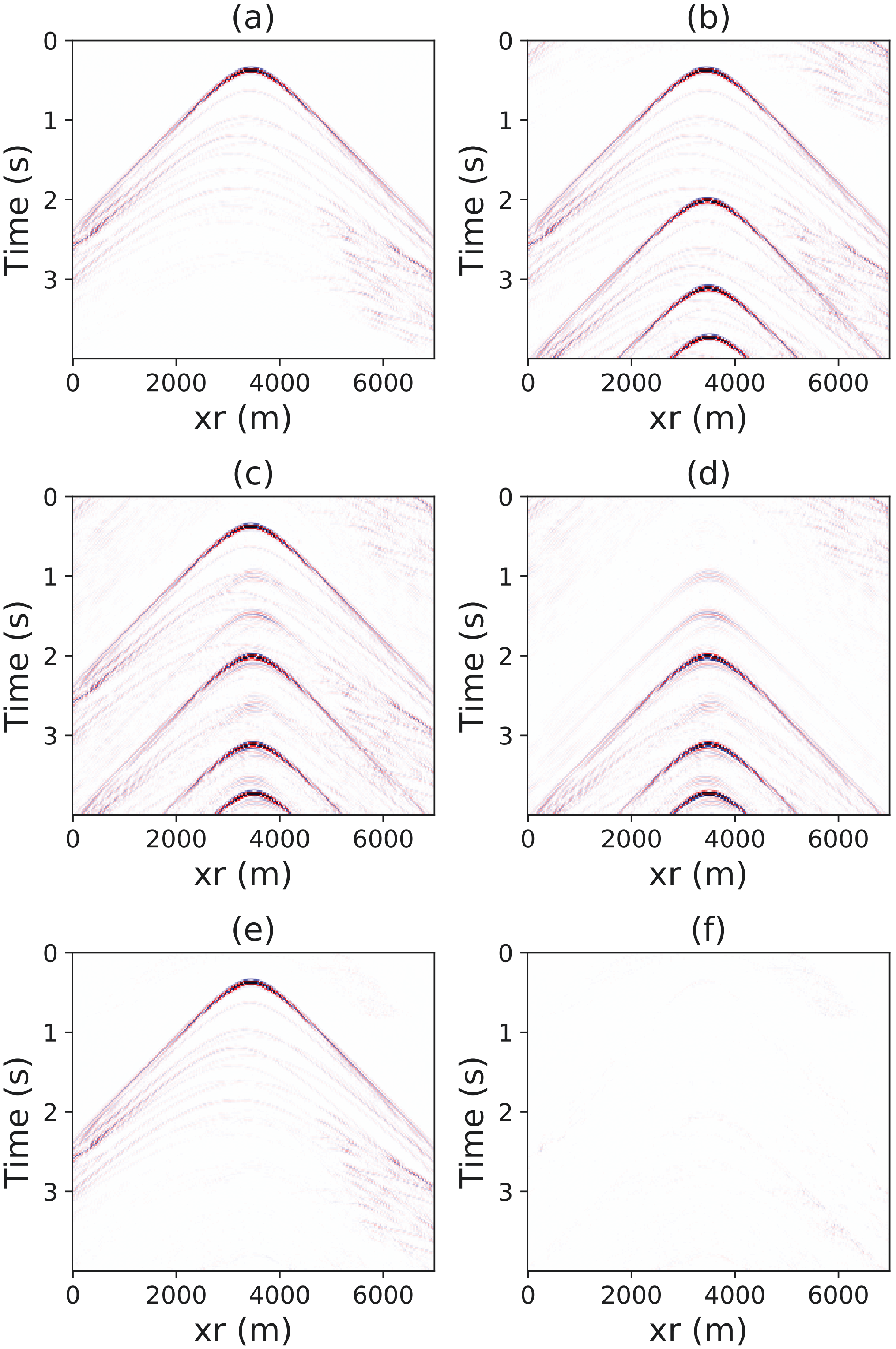

Figure 3.10: (a) One ideal common shot gather. (b) Pseudo-deblended data shot receiver gather $SNR_{in} = -3.6$ dB. (c) Deblending via classical Matching Pursuit $SNR_{out} = -3.4$ dB. (d) Difference between (a) and (c). (e) Deblending via robust Matching Pursuit $SNR_{out} = 18.8$ dB. (f) Difference between (a) and (e).

from the Gulf of Mexico with a total of 808 shots and 183 receivers. The receivers are evenly distributed with an interval of 87 m. We blend three consecutive sources to perform pseudo-blending. In this example, we apply deblending in the common channel gather domain first and then sort back the data to common shot gathers. As in the previous case, we adopted a local linear Radon transform on overlapping windows of 10 traces and 400 time samples.

Figure 3.11 shows one ideal common offset gather and its associated pseudo-deblended gather. As expected, dips in common offset gathers change laterally. Hence, one has to adopt a local denoising strategy, which corresponds to applying linear Radon transform denoising in windows. Figures 3.11c and 3.11e shows the result for classical Matching Pursuit and Robust Matching Pursuit. For the Robust Matching pursuit algorithm, the SNR improves from $SNR_{in} = -0.65$ dB to $SNR_{out} = 11.7$ dB. The classical Matching Pursuit algorithm was not able to attenuate the blending noise.

Similar to the 2D finite differences example, we apply deblending to all common offset gathers, and then sort them back into common shot gathers. Figure 3.12 corresponds to one of the common shot gathers and its pseudo-deblended data. The pseudo-deblended data contains source interference from two neighboring sources ($SNR_{in} = -1.85$ dB). Figure 3.12e shows deblended data obtained via the robust Matching Pursuit algorithm. In this case, the signal-to-noise ratio increases from $SNR_{in} = -1.85$ dB to $SNR_{out} = 13.2$ dB. The quality enhancement can also be observed even from one single trace, as it is portrayed in Figure 3.13.

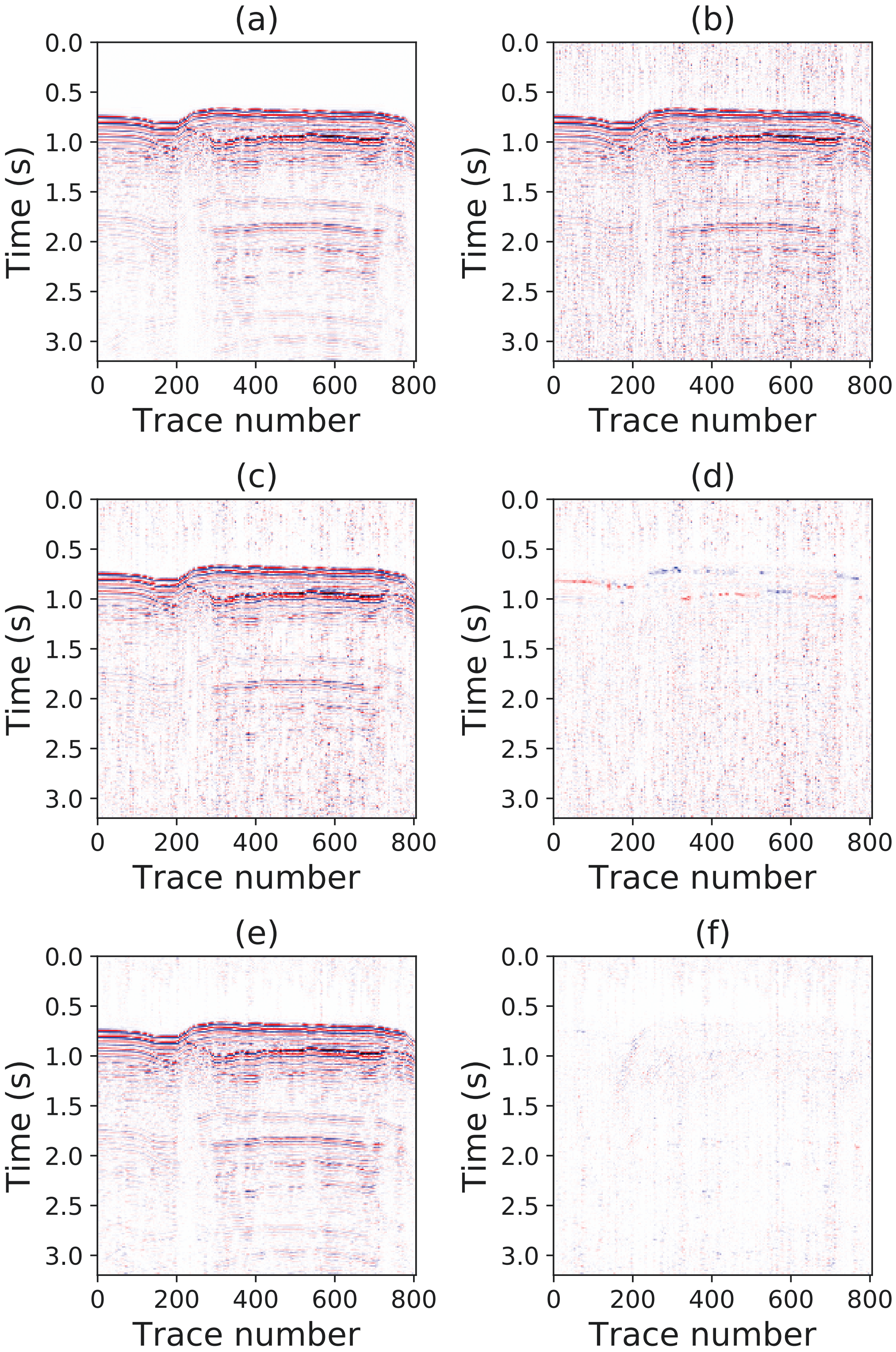

Figure 3.11: (a) One ideal common-offset section. (b) Pseudo-deblended common offset section $SNR_{in} = -0.65$ dB. (c) Non-robust deblending $SNR_{out} = 2.7$ dB. (d) Difference between (a) and (c), (a)-(c). (e) Robust deblending $SNR_{out} = 11.7$ dB. (f) Difference between (a) and (e).

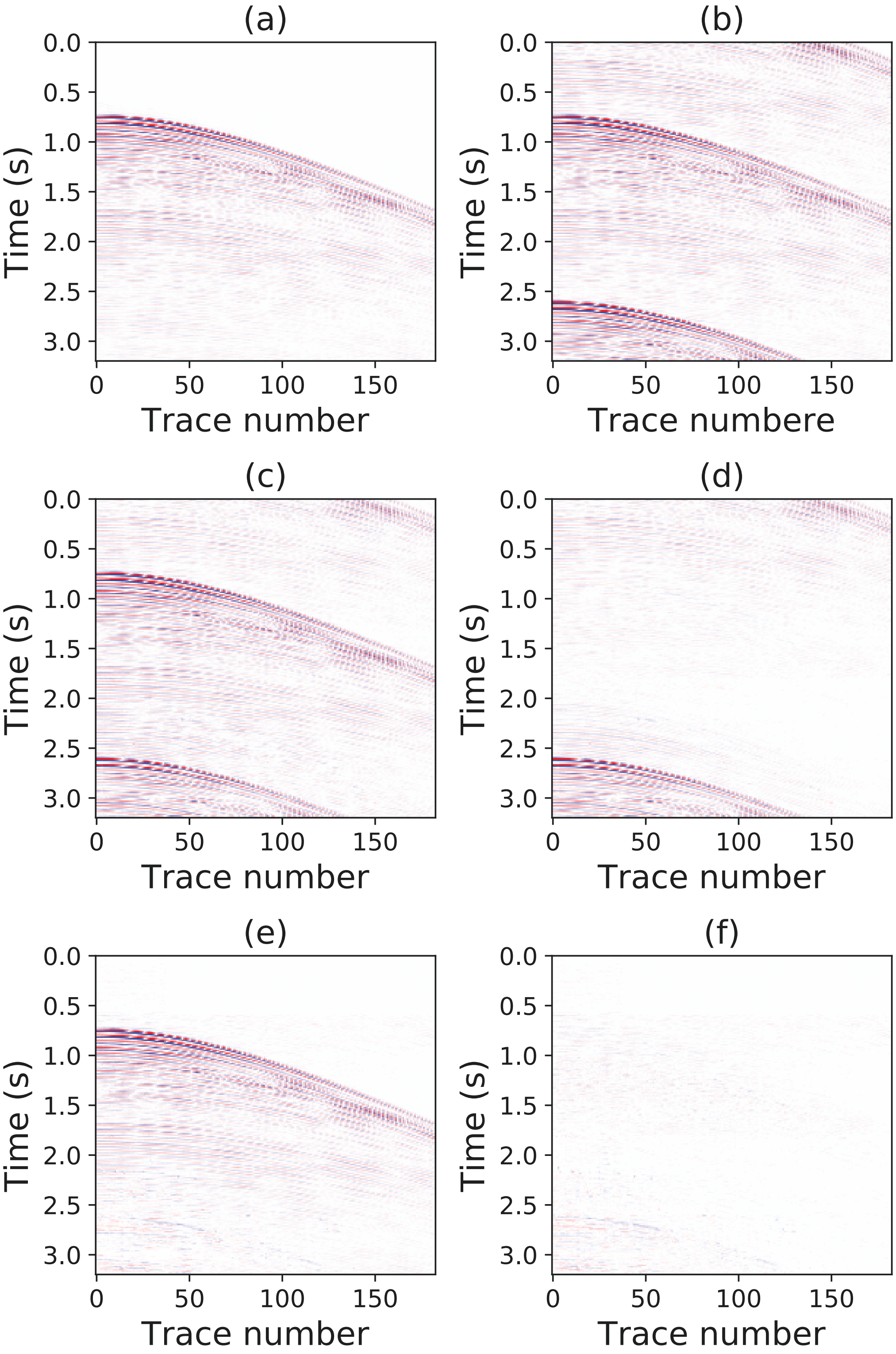

Figure 3.12: Deblending results for a commom shot gather for the real marine dataset. (a) Ideal shot gather. (b) Pseudo-deblended data with $SNR_{in} = -1.85$ dB. (c) Non-robust deblending result with $SNR_{out} = 1.7$ dB. (d) Difference between (a) and (c). (e) Robust deblending result $SNR_{out} = 13.2$ dB. (f) Difference between (a) and (e).

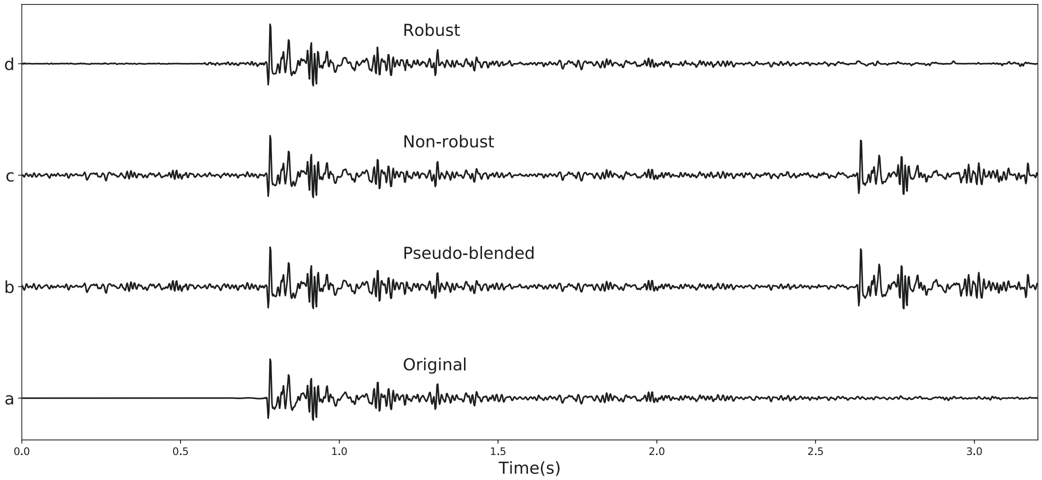

Figure 3.13: Trace-by-trace comparison for a near offset trace. (a) Ideal seismic trace. (b) Trace after pseudo-deblending. (c) Denoising after classical Matching Pursuit. (d) Denoising after robust Matching Pursuit.

3.3 Discussion

We have tested the robust Matching Pursuit algorithm for different incoherent noise levels and with diverse synthetic datasets. In general, the algorithm is not sensitive to selecting the parameter p as far as $0.5 < p < 1.2$. In our examples, we choose $p = 0.8$ because the value provided a slightly better result than for other values of p. Additional parameters for consideration are the maximum number of iterations and the tolerance η, which monitors the residuals' decay versus iteration. For the finite-difference and real data examples, we select more than one coefficients in each interaction to reduce the number of total iterations and hence minimize computational cost. Using many iterations for the non-robust Matching Pursuit algorithm only leads to overfitting the erratic blending noise at the expense of choosing the wrong basis functions. On the other hand, the robust Matching Pursuit algorithm can counterbalance the influence of erratic blending noise. Hence, in each iteration, the robust Matching Pursuit method is less likely to select incorrect basis functions.

We have also explored early stopping strategies for the classical non-robust Matching Pursuit to examine if satisfactory deblending results were attainable by this mean. Unfortunately, our tests show that classical Matching Pursuit cannot cope with erratic impulsive noise introduced by source interferences in common receiver and common channel gathers. This is a consequence of the classical Matching Pursuit formulation, which was developed under assumptions that do not contemplate erratic noise.

We recognize that many other methods can be used to solve the deblending problem. To mention a few options, one can adopt methods based on Radon transforms (Moore et al., 2008; Haacke et al., 2015), Fourier operators (Abma et al., 2010), the Seislet transform (Chen et al., 2014; Gan et al., 2016), rank-reduction (Cheng and Sacchi, 2015; Jeong et al., 2020), curvelet transform (Kontakis and Verschuur, 2017), etc. Comparing the proposed algorithm to the plethora of algorithms that have been proposed for simultaneous source separation is not the intention of our paper. Our main goal is to present a robust Matching Pursuit algorithm for computing a Radon denoiser and, in particular, to show how it can be adopted for processing simultaneous source data. We also stress that we provide a detailed description that permits to implement robust Matching Pursuit with an implicit operator such as the time-domain Radon transform. A similar strategy can also be adopted for other operators that are applied in implicit form. An example of the latter is multidimensional Fourier interpolation, where forward and adjoint operators are Fast Fourier Transforms (Liu and Sacchi, 2004; Trad, 2009) or non-uniform Fast Fourier Transforms (Xu et al., 2005; Zwartjes and Sacchi, 2007; Schonewille et al., 2013).

CHAPTER 4

Fast Robust Greedy methods

In the last chapter, I introduced the traditional Matching pursuit algorithm (Mallat and Zhang, 1993). I also combined it with the robust inner product method and proposed a new robust Matching pursuit method. I then applied the robust MP to the simultaneous source separation problem. The robust MP gives a decent deblending performance. However, one major drawback of the traditional MP is its computational costs when applied to large-scale datasets. MP only selects one atom from the dictionary D in each iteration, and it will also generally repeatedly pick the same atom from D to refine the approximation further. The algorithm is guaranteed to stop in a finite number of iterations if the norm of residual $\mathbf{r}$ is used to define the stopping criteria. For sparse data with k non-zeros coefficients, the MP requires at least k iterations for full reconstruction. When the number k is large, this process can be impractically slow. From the previous examples, and in particular for the complex finite different example and real marine data example, hundreds of thousands of non-zero coefficients need to be reconstructed in the Radon domain. Hence, the robust MP takes a considerable amount of time to deblend the full dataset. In this chapter, I will introduce a fast greedy pursuit algorithm. I will compare the performance of different algorithms for noise-free, random noise, and blending noise cases. I will also explain how to apply the robust inner product, and robust Radon transforms to these fast greedy pursuit algorithms. In the end, I will use the fast robust greedy method to run the deblending problem for the finite different example and real marine example again and compare the results with the robust MP we proposed in chapter 3.

4.1 Fast Greedy Pursuit

4.1.1 Orthogonal Matching Pursuit

We first need to introduce the orthogonal Matching Pursuit (OMP) (Tropp and Gilbert, 2007; Tropp, 2004) before introducing other fast greedy pursuit algorithms since all of them are based on the OMP scheme. Unlike MP, which updates one coefficient in each iteration, OMP minimizes the following cost function for all of the currently selected coefficients in each iteration

$$\widehat{\mathbf{x}}^{[i]}_{T^{[i]}} = \underset{\widetilde{\mathbf{x}}_{T^{[i]}}}{\operatorname{argmin}} \ \|\mathbf{y} - \mathbf{A}_{T^{[i]}} \widetilde{\mathbf{x}}_{T^{[i]}} \|_2^2, \tag{4.1}$$

where $T^{[i]}$ is the set of the indexes of all coefficients we have been picked until iteration i. The full algorithm for the OMP is listed below. Since we update all currently selected coefficients in each iteration, unlike MP, OMP will never re-select the atoms. Moreover, the residual vector in each iteration is always orthogonal to all currently selected atoms.

However, although OMP never re-selects the atom, it still needs k iterations to fully reconstruct the k-sparse problem. Also, the orthogonalization step is usually the

Algorithm 3 OMP

```
Input: $\mathbf{y}$, $\mathbf{A}$ ,and $k$
Output: $\mathbf{r}^{[k]}$, $\widehat{\mathbf{x}}^{[k]}$
Initialization: $\mathbf{r}^{[0]} = \mathbf{y}$ ,$\widehat{\mathbf{x}}^{[k]} = 0$, and $T^{[0]} = \emptyset$
for $k = 1, 2, ...., K$ do
    $l = \underset{j=1,2,....M}{\text{argmax}} \; |\langle \mathbf{A}, \mathbf{r}^{[k-1]} \rangle|$
    $T^{[k]} = T^{[k-1]} \cup \{l\}$
    $\widehat{\mathbf{x}}^{[k]}_{T^{[k]}} = \underset{\widetilde{\mathbf{x}}_{T^{[k]}}}{\text{argmin}} \; \|\mathbf{y} - \mathbf{A}_{T^{[k]}} \widetilde{\mathbf{x}}_{T^{[k]}} \|_2^2$
    $\mathbf{r}^{[k]} = \mathbf{y} - \mathbf{A}\widehat{\mathbf{x}}^{[k]}_{T^{[k]}}$
end for
```

bottleneck, especially for large-scale data. From the algorithm above, we find the main cost of OMP is coming from two parts. Every time we calculate the inner product between the residual vector and a basis vector, the other is the orthogonalization step where one needs to minimize an $l_2 - norm$ cost function. Thus, two realistic strategies can be used to accelerate the OMP algorithm. The first is to pick more than one coefficient in each iteration to reduce the total time to calculate inner products. Secondly, one can modify the orthogonalization minimization part to reduce computations further. I, therefore, will discuss both of these two strategies in next sections.

4.1.2 Selection strategies

We first discuss selection strategies which can pick more than one element in each iteration.

StOMP

One of the approaches that uses the thresholding idea for multi-element selection in each iteration is the Stagewise Orthogonal Matching Pursuit(StOMP) (Donoho et al., 2012). In this approach, a threshold λ is calculated depending on the current residual $\mathbf{r}$

$$\lambda_{stomp} = \frac{t\|\mathbf{r}^{[k-1]}\|_2}{\sqrt{M}}, \tag{4.2}$$

where M is the dimension of $\mathbf{x}$, and $\frac{\|\mathbf{r}^{[k-1]}\|_2}{\sqrt{M}} = \sigma$ represent the noise level and t is a threshold parameter. The set of indexes is then updated as

$$T^{[k]} = T^{[k-1]} \cup \{i : |g_i| \geq \lambda_{stomp}\}. \tag{4.3}$$

The performance of StOMP is only guaranteed for some specific matrix $\mathbf{A}$. Moreover, theoretical performance guarantees are not available when applied to more general matrices. From the equation giving the thresholding value λ, one finds that selecting the parameter t is critical for its performance. In fact, there are no intuitive guidelines available to select t, and the only suggestion in (Donoho et al., 2012) is to use a value between 2 and 3. Another major problem is that one only uses the residual $\mathbf{r}$ to define the threshold. The algorithm might be terminated when all the inner products fall below the threshold. This problem is more probable when the algorithm is applied to data with erratic noise since the erratic noise remains in the residuals and makes them relatively large. In the end, this approach shows mixed results.

Regularized OMP

Another multi-element selection strategy that has been proposed is the Regularized OMP (ROMP) (Needell and Vershynin, 2008; Needell and Vershynin, 2010). The ROMP algorithm groups the inner products $\mathbf{g}_i$ into sets J_k such that the elements in each set have a similar magnitude,

$$|\mathbf{g}_i| \leq \frac{1}{r}|\mathbf{g}_j|, \quad \text{for all} \quad i, j \in J_k. \tag{4.4}$$

ROMP then selects the set J_k for which $\|\mathbf{g}_{J_k}\|_2$ is largest.

For the ROMP selection strategy proposed in (Needell and Vershynin, 2008; Needell and Vershynin, 2010), r was assumed to be 0.5. Theoretically, the ROMP performance should be better than the performance of OMP. However, in many practical situations, ROMP's average performance was notably worse than that of OMP and StOMP. Also, the grouping and selection method is much more complex compare with other multi-element selection methods. Therefore, ROMP is not considered a good practical algorithm for CS or sparse data reconstruction and it will not be tested in the following sections.

SWOMP

Both ROMP and StOMP have several drawbacks. Luckily, there is another multi-element selection strategy that has been proposed by Blumensath and Davies (2009), which is called stagewise weak OMP (SWOMP). The idea of SWOMP is inspired by the weak matching pursuit algorithm (Mallat, 2008). Weak matching pursuit is a method developed for large or infinite-dimensional problems in which not all inner products can be evaluated explicitly. For the stagewise weak selection, we select all elements that satisfy the following update condition

$$T^{[k]} = T^{[k-1]} \cup \{i : |\mathbf{g}_i| \geq \alpha \text{max} |\mathbf{g}_j|\}. \tag{4.5}$$

It will select all elements that come within a factor of α of the largest inner product (in magnitude), and $\alpha \in (0, 1]$. It has been proved that SWOMP is preferable in many respects to the other two methods.

4.1.3 Coefficient updates

The second approach to reducing the total cost is by modifying the coefficient optimization step. This step is done using an orthogonal projection traditionally. However, for applications in which $\mathbf{A}$ is large, it will be problematic both in storage and computational cost.

Conjugate Gradient Pursuit

Traditionally, in each iteration, we start a new conjugate gradient solver every time and fully run it, and it gives us the optimal result in each iteration. In order to save the total cost, one can use few iterations of conjugate gradient and get the suboptimal result in each iteration like what Donoho does in his paper about Stagewise Orthogonal Matching Pursuit (StOMP) (Donoho et al., 2012). StOMP uses a small number of conjugate gradient steps in each iteration to reach a approximate orthogonalization. In the article introducing SWOMP (Blumensath and Davies, 2009), the author also introduces a fast way to perform the coefficient update step, called conjugate gradient pursuit (CGP) which is belongs to the Gradient Pursuit framework (GP) (Blumensath and Davies, 2008a). The Gradient Pursuit algorithm use directional optimisatation to update the data $\hat{\mathbf{d}}^{[k-1]}$ in each iteration, the update for MP and OMP can also works in this way. So for the MP and OMP, the update directions for the approximate conjugate gradient method can be calculated using

$$\mathbf{d}^{[i]}_{T^{[i]}} = \mathbf{g}^{[i]}_{T^{[i]}} + \nu \mathbf{d}^{[i]}_{T^{[i-1]}}. \tag{4.6}$$

where $\nu^{[i]} = \frac{\langle (\mathbf{A}_{T^{[i-1]}}\mathbf{d}^{[i-1]}_{T^{[i-1]}}),(\mathbf{A}_{T^{[i]}}\mathbf{g}^{[i]}_{T^{[i]}})\rangle}{\|\mathbf{A}_{T^{[i-1]}}\mathbf{d}^{[i-1]}_{T^{[i-1]}}\|_2^2}$ ensures that $\langle \mathbf{A}_{T^{[i]}}\mathbf{d}^{[i]}_{T^{[i]}}, \mathbf{A}_{T^{[i]}}\mathbf{d}^{[i-1]}_{T^{[i]}}\rangle = 0$. It is important to note that this strategy uses a single update direction after each element selection step. This update direction is chosen to be conjugate to the update step in the previous iteration. Therefore, instead of starting a new conjugate gradient and fully running it in each iteration as OMP, the CGP only runs a conjugate gradient one time in each iteration. The general algorithm of CGP combines with OMP with different selection strategies can be summarized as bellow.

Algorithm 4 CGP

```
Input: $\mathbf{y}$, $\mathbf{A}$ ,and $k$
Output: $\mathbf{r}^{[k]}$, $\widehat{\mathbf{x}}^{[k]}$
Initialization: $\mathbf{r}^{[0]} = \mathbf{y}$ ,$\widehat{\mathbf{x}}^{[k]} = 0$, and $T^{[0]} = \emptyset$
for $k = 1, 2, ...., K$ do
    $\mathbf{g}^{[k]} = \mathbf{A}^T \mathbf{r}^{[k-1]}$
    Select a set of new elements $\mathcal{I}$.
    $T^{[k]} = T^{[k-1]} \cup \mathcal{I}$
    if n=1 then
        $\mathbf{d}^{[k]}_{T^{[k]}} = \mathbf{g}^{[k]}_{T^{[k]}}$
        $\mathbf{v}^{[k]} = \mathbf{A}_{T^{[k]}} \mathbf{d}^{[k]}_{T^{[k]}}$
    else
        $\mathbf{w}^{[k]} = \mathbf{A}_{T^{[k]}} \mathbf{g}^{[k]}_{T^{[k]}}$
        $\nu^{[k]} = -\langle \mathbf{v}^{[k-1]}, \mathbf{w}^{[k]}\rangle / \eta^{[k-1]}$
        $\mathbf{d}^{[k]}_{T^{[k]}} = \mathbf{g}^{[k]}_{T^{[k]}} + \nu^{[k]} \mathbf{d}^{[k-1]}_{T^{[k]}}$
        $\mathbf{v}^{[k]} = \mathbf{w}^{[k]} + \nu^{[k]} \mathbf{v}^{[k-1]}$
    end if
    $\eta^{[k]} = \|\mathbf{v}^{[k]}\|_2^2$
    $\beta^{[k]} = \langle \mathbf{r}^{[k-1]}, \mathbf{v}^{[k]}\rangle / \eta^{[k]}$
    $\mathbf{x}^{[k]}_{T^{[k]}} = \mathbf{x}^{[k-1]}_{T^{[k]}} + \beta^{[k]} \mathbf{d}^{[k]}_{T^{[k]}}$
    $\mathbf{r}^{[n]} = \mathbf{r}^{[n-1]} - \beta^{[n]} \mathbf{v}^{[n]}$
    $n \leftarrow n + 1$
end for
```

Theoretically, since we are not entirely run the conjugate gradient in the algorithm, the orthogonalization in each iteration is sub-optimal and approximated. The conjugate gradient pursuit will re-select the same atoms. If the residual is far from orthogonal, the inner product with already selected elements will be large, and the elements will be re-selected. If not, CGP will not re-select the same elements(Blumensath and Davies, 2009) like MP. Therefore, the final performance will not be affected. Overall, the conjugate gradient pursuit has a similar performance as OMP. The computational cost in each iteration is close to the MP, and combine with the selection strategies like stagewise weak selection it can reduce the total iterations needed a lot. In the next section, I will use simple tests to compare the performance of different selection strategies.

4.1.4 Examples

The first test is done with noise-free data. The linear Radon model was used in Figure 4.1 to generate a model that consists of four linear events. The coefficients in Radon space are the unknown of our problem. Notice, these coefficients where convolved with a Ricker wavelet with a central frequency of 25 Hz. There is a total of 92 non-zero coefficients in the linear Radon panel in Figure 4.1a. Figure 4.1b is the synthetic data generated by the Radon coefficients via forward modelling with the linear Radon transform.

We set the total number of iterations to $k = 100$ for all algorithms. For StOMP, we use $t = 100$, and $\alpha = 0.7$ for SWOMP. Figure 4.2 shows the results of the convergence for different algorithms. I also need to mention that for all OMP, StOMP, and SWOMP, we use the conjugate gradient pursuit mentioned above to update the coefficients. Figure 4.2 shows that MP needs the most iterations to reconstruct the data entirely, and OMP needs fewer iterations than MP. For both StOMP and

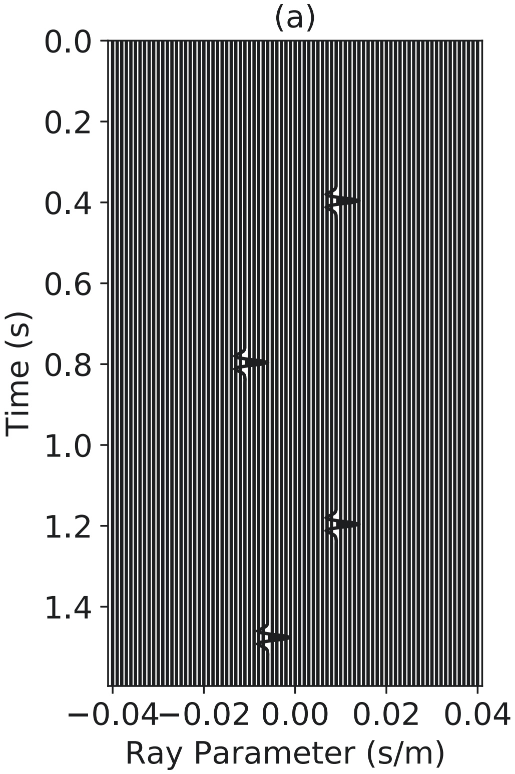

Figure 4.1: (a) The linear Radon used to generate the synthetic data. (b) Synthetic data generated from (a) with a forward linear Radon operator.

SWOMP have converged much faster than the traditional OMP algorithm. We can also stop the algorithm when the total number of the non-zero selected coefficients meets a desired number. There are 92 non-zero coefficients in our original Radon panel. Thus we set the desired total coefficients of $n = 92$. Our results show that it takes 246 iterations for MP to pick the 92 coefficients, as we know the MP tends to re-select coefficients for further optimization. OMP takes exactly 92 iterations to stop the algorithm. For StOMP with $t = 100$, it only needs 6 iteration, and SWOMP with $\alpha = 0.7$ needs 11 iterations. Therefore for the noise-free data, both StOMP and SWOMP work perfectly.

Figure 4.3 shows the convergence results of the StOMP method with different values

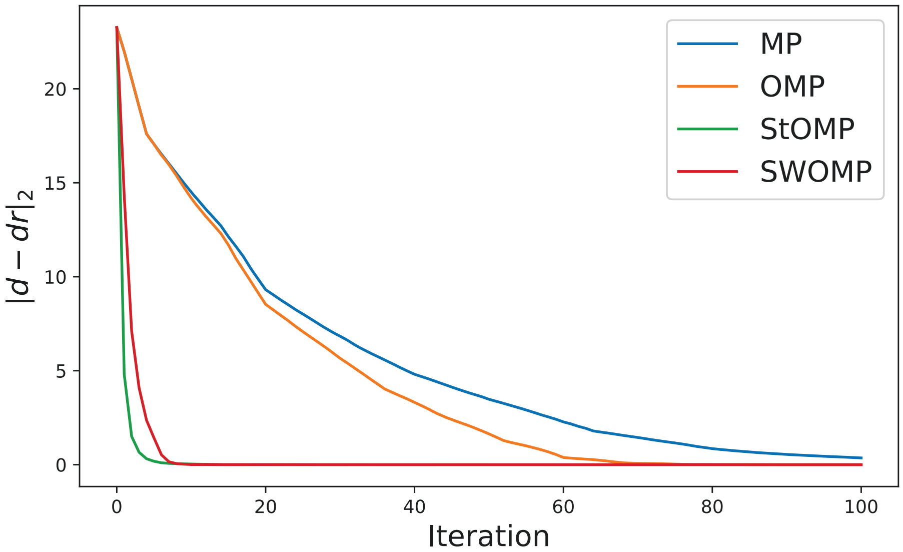

Figure 4.2: Comparison of convergence with different algorithms for the noise-free case. The x-axis is the iteration number and the y-axis is the norm of the residual $\mathbf{r}$.

of t. When we use $t = 250$ and $t = 260$, StOMP works very well, but when we use $t = 270$ and $t = 280$, the algorithm failed because all inner products fall below the threshold after several iterations. Although the wrong value of t can make StOMP fail, it is still relatively easy to find the value of t that works well in the noise-free case. Figure 4.4 shows convergence results of SWOMP for different values of α. It shows that the value of α is only affecting the convergence rate of the algorithm but not the final denoising performance. I also need to point out that when $\alpha = 1$, the SWOMP is equivalent to OMP.

Next, we test the performance of these multi-element selection strategies with data contaminated with random noise. The noisy data are presented in Figure 4.5 b. Figure 4.5 shows the final results. This time, StOMP failed to reconstruct the noisy data fully. All MP, OMP, and SWOMP still work well. Unlike before, since we have

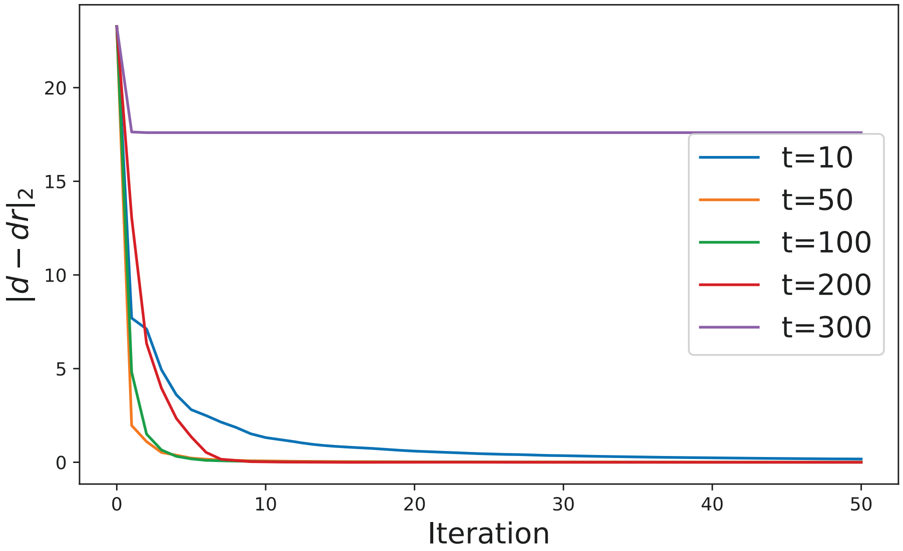

Figure 4.3: Comparison of the convergence of StOMP for different values of t for the noise-free case. The x-axis is the iteration number and the y-axis is the norm of the residual $\mathbf{r}$.

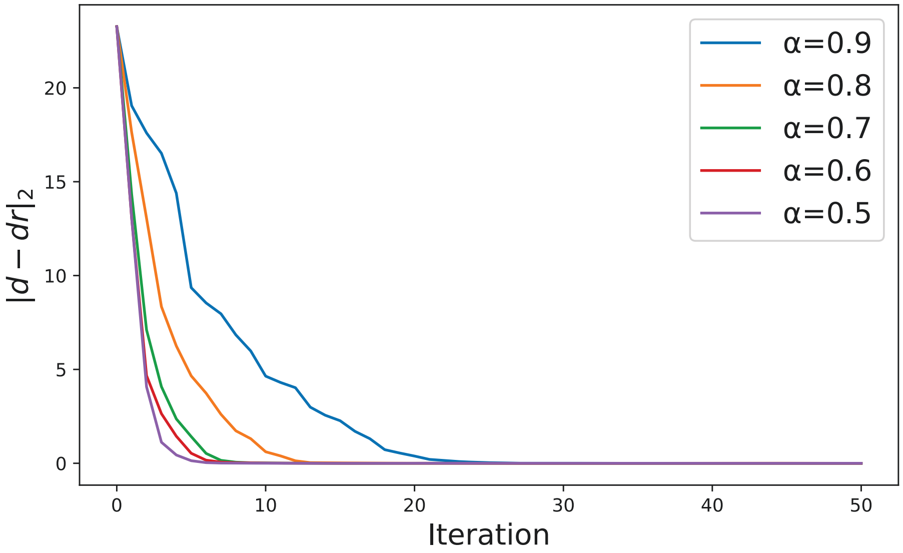

Figure 4.4: Comparison of the convergence of SWOMP for different values of α for the noise-free case. The x-axis is the iteration number and the y-axis is the norm of the residual $\mathbf{r}$.

random noise in our data, we stopped our algorithms when the desired number of non-zero coefficients were picked. This time, we set the non-zero coefficients we want to pick to $n = 60$.

Figure 4.7 shows the convergence results for StOMP with different value of t. When we use a small value of t, the algorithm picked many coefficients in the very early iterations, and it stopped after a few iterations. If one slightly increases the value of t, all the inner product became smaller than the threshold λ and the algorithm terminates after few iterations again. We can probably find an optimal value of t by doing many tests, which can make the algorithm work again for the noisy data . StOMP is not a good option for the noisy data case. Figure 4.8 shows the same results as before. For the noisy data, the parameter α does not change the final denoising performance of the SWOMP method.

4.2 Robust fast Greedy methods

I tested greedy inversion algorithms with different coefficient selection strategies for both noise-free case and random noise case. I will apply the robust inner product and robust Radon transform introduced in Chapter 3 to the algorithm presented in this chapter. I will also test the performance of these different selection strategies again for data contaminated with erratic noise. The implement of robust inner product and robust Radon transform into these algorithms is straightforward. We use the robust inner product to calculate the robust adjoint radon coefficients $\mathbf{g}^{[k]}$ in each iteration. To make the problem robust to erratic noise, I replace all the traditional inner products by a robust inner product in ℓ_p space in the coefficient optimization part. I have already explained how to apply the Radon transform with ℓ_p inner products in Chapter 3 and it will not be explained here again.The following

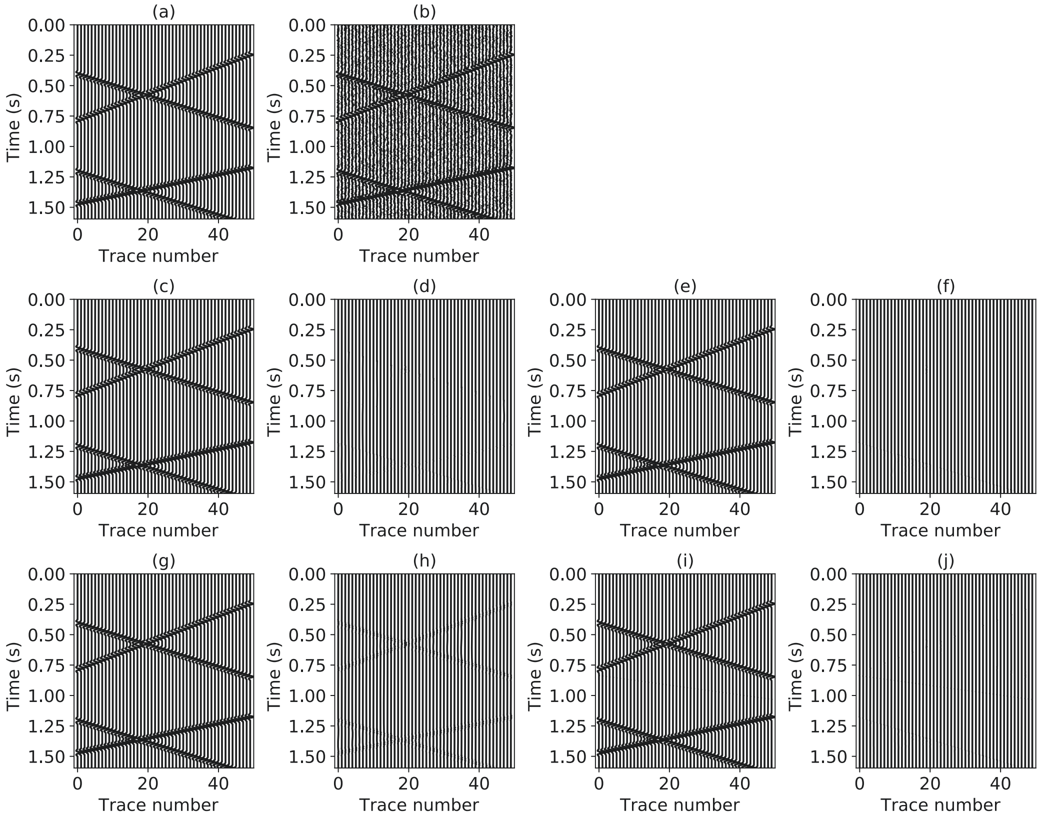

Figure 4.5: (a) Original data. (b) Noisy data. (c) Denoising result by RMP, $SNR = 15.4$ dB. (d) Error between a and c. (e) Denoising by ROMP, $SNR = 16.6$ dB. (f) Error between a and e. (g) Denoising by RStOMP, $SNR = 9.8$ dB. (h) Error between a and g. (i) Denoising by RSWOMP, $SNR = 16.4$ dB. (j) Error between a and i.

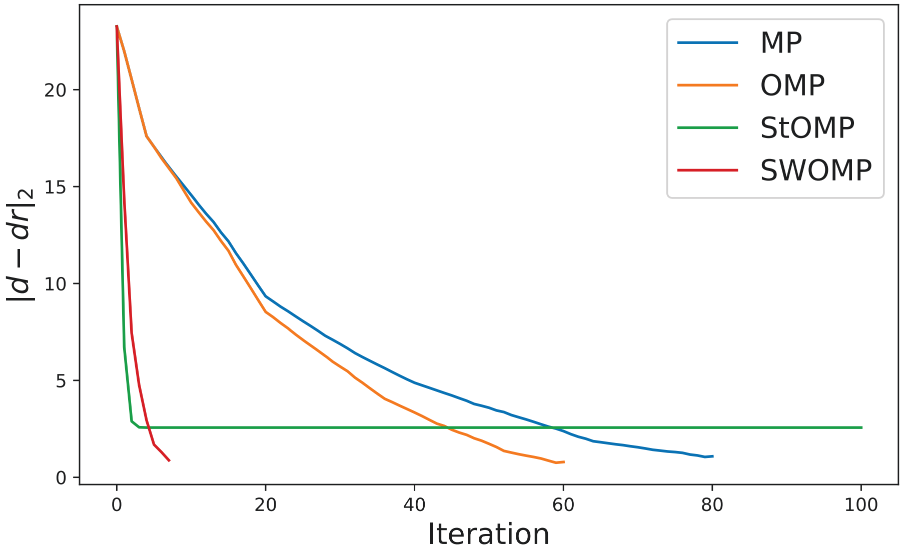

Figure 4.6: Comparison of the convergence with different algorithms for data contaminated with random noise. The x-axis is the iteration number and the y-axis is the norm of the residual **r**.

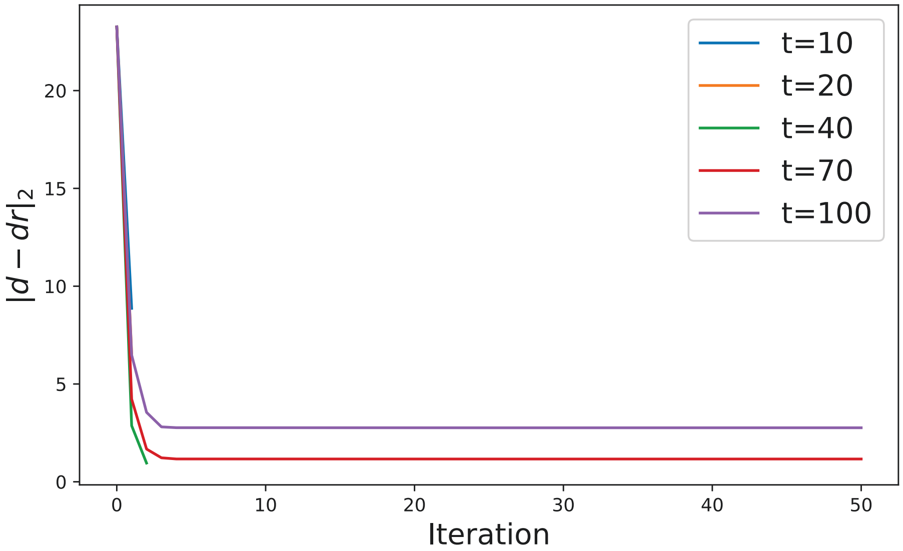

Figure 4.7: Comparison of the convergence of StOMP for different values of t for data contaminated with random noise. The x-axis is the iteration number and the y-axis is the norm of the residual **r**.

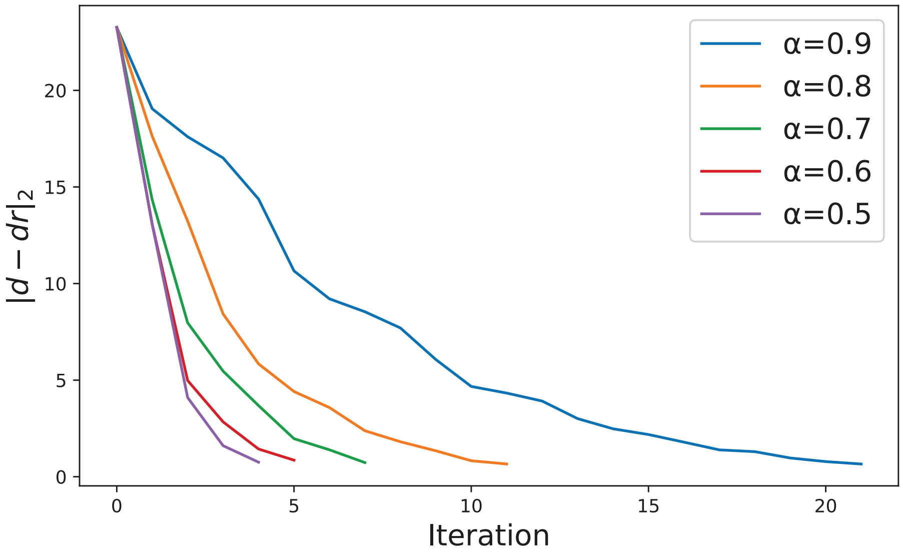

Figure 4.8: Comparison of the convergence of SWOMP for different values of α for the data contaminated with random noise. The x-axis is the iteration number and the y-axis is the norm of the residual $\mathbf{r}$.

is the general algorithm for the robust CGP with different selection methods and the robust Radon operator.

Where L is the forward Radon operator. We can stop our algorithm when the maximum iterations are reached, or the relative change of residual r becomes very small.

4.2.1 Simple example

I test the denoising performance of the robust CGP with the same simple example used previously. The synthetic data with erratic noise to represent the common receiver gather presented in Figure 4.9 b. I use a blending factor equal to 4 to generate this pseudo-deblended common receiver gather. As before, to avoid overfitting, I stop the algorithms when 60 non-zero coefficients are picked. The robust

Algorithm 5 RCGP

```
Input: $\mathbf{y}$, $L$ ,and $k$
Output: $\mathbf{r}^{[k]}$, $\widehat{\mathbf{x}}^{[k]}$
Initialization: $\mathbf{r}^{[0]} = \mathbf{y}$ ,$\widehat{\mathbf{x}}^{[k]} = 0$, and $T^{[0]} = \emptyset$
for $k = 1, 2, ...., K$ do
    for  all $\tau, q$ do
        $\mathbf{u}(\tau, q) = [d(\tau + q\,\phi(h_1), h_1), d(\tau + q\,\phi(h_2), h_2), \ldots, d(\tau + q\,\phi(h_{n_h}), h_{n_h})]^T$
        $\mathbf{g}^{[k]}(\tau, q) = (\mathbf{u}(\tau, q), \mathbf{1})_p$
    end for
    Select a set of new elements   $\mathcal{I}$.
    $T^{[k]} = T^{[k-1]} \cup \mathcal{I}$
    if n=1 then
        $\mathbf{d}^{[k]}_{T^{[k]}} = \mathbf{g}^{[k]}_{T^{[k]}}$
        $\mathbf{v}^{[k]} = L\mathbf{d}^{[k]}_{T^{[k]}}$
    else
        $\mathbf{w}^{[k]} = L\mathbf{g}^{[k]}_{T^{[k]}}$
        $\nu^{[k]} = -\text{LP-norm}(\mathbf{v}^{[k-1]}, \mathbf{w}^{[k]}, lp)$
        $\mathbf{d}^{[k]}_{T^{[k]}} = \mathbf{g}^{[k]}_{T^{[k]}} + \nu^{[k]}\mathbf{d}^{[k-1]}_{T^{[k]}}$
        $\mathbf{v}^{[k]} = \mathbf{w}^{[k]} + \nu^{[k]}\mathbf{v}^{[k-1]}$
    end if
    $\beta^{[k]} = \text{LP-norm}(\mathbf{r}^{[k-1]}, \mathbf{v}^{[k]}, lp)$
    $\mathbf{x}^{[k]}_{T^{[k]}} = \mathbf{x}^{[k-1]}_{T^{[k]}} + \beta^{[k]}\mathbf{d}^{[k]}_{T^{[k]}}$
    $\mathbf{r}^{[n]} = \mathbf{r}^{[n-1]} - \beta^{[n]}\mathbf{v}^{[n]}$
    $n \leftarrow n + 1$
end for
```

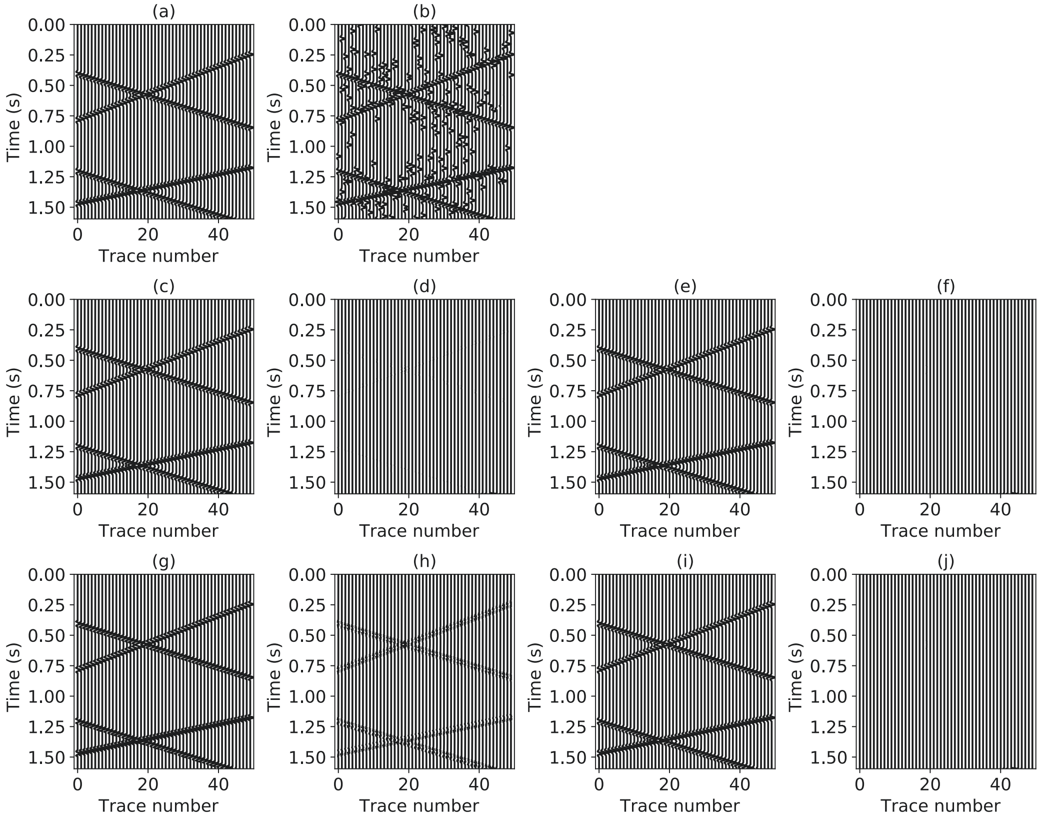

Figure 4.9: (a) Original data. (b) Pseudo-blended data. (c) Denoising result by RMP, SNR=13.4 dB. (d) Error between a and c. (e) Denoising by ROMP, SNR=13.4 dB. (f) Error between a and e. (g) Denoising by RStOMP, SNR=6.6 dB. (h) Error between a and g. (i) Denoising by RSWOMP, SNR=13.4 dB. (j) Error between a and i.

StOMP failed again for the erratic noise case. Robust OMP and SWOMP have the best performance. From figure 4.11 , I find that for the erratic noise case, even a slight change of the parameter t can change the performance of the algorithm dramatically. It is almost impossible to find a value of t that works for this example. The test for the robust SWOMP shows the same results as before, the value of the parameter α only affect the convergence rate but not the denoising performance.

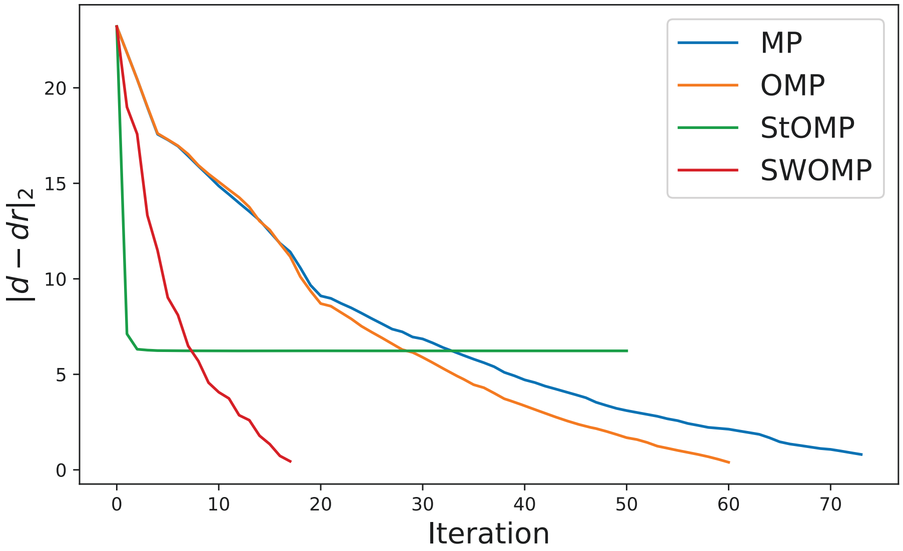

Figure 4.10: Comparison of convergence with different algorithms for the blending noise case. The x-axis is the iteration number and the y-axis is the norm of the residual **r**.

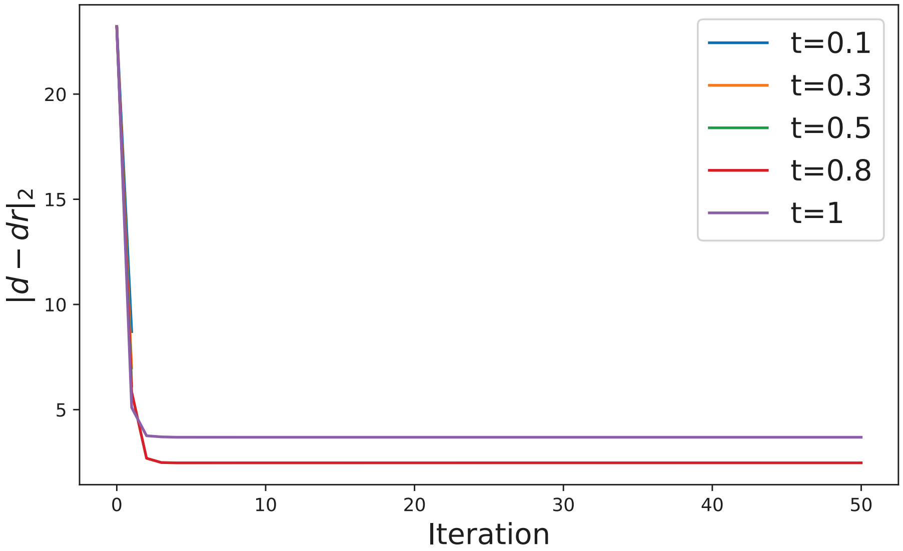

Figure 4.11: Comparison of convergence of StOMP with different value of t for the blending noise case. The x-axis is the iteration number and the y-axis is the norm of the residual **r**.

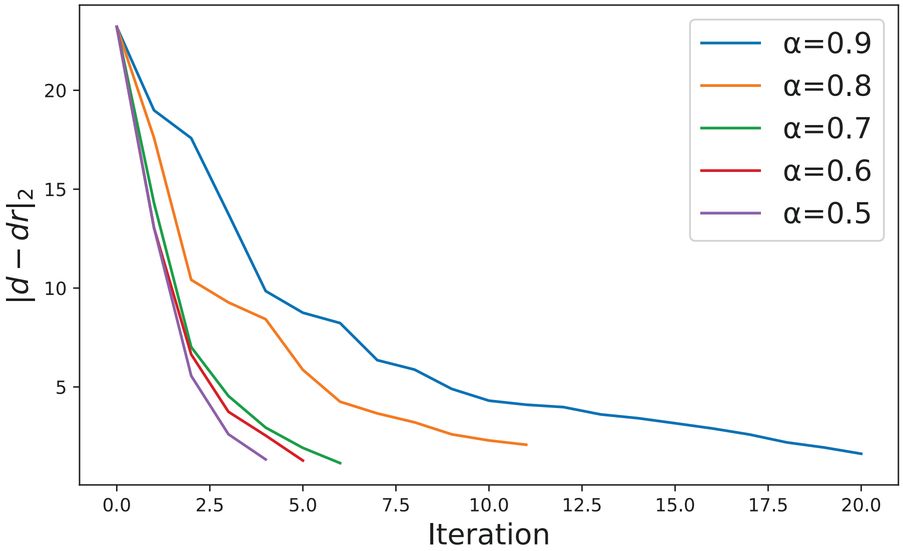

Figure 4.12: Comparison of convergence of SWOMP with different value of α for the blending noise case. The x-axis is the iteration number and the y-axis is the norm of the residual $\mathbf{r}$.

4.3 Complex examples

In this section, I re-do the deblending example for the complex finite different example and real marine data example and solve the problem via the fast robust greedy pursuit algorithm. After the previous tests, I decided to use the Stagewise Weak Orthogonal Matching Pursuit as the selection strategy for the fast robust greedy pursuit. I am following the same deblending scheme explained in Chapter 3 and re-do the deblending process for the finite different example and real marine data example.

Figure 4.13 and 4.14 show the deblending result for the finite different data example in both common receive gather and common shot gather. Similarly, Figure 4.15 and 4.16 show the deblending result for the real marine data example in both common offset gather and common shot gather. For these tests, we set the maximum iteration for each window equals to 20. From these results, we can conclude that robust Stagewise Weak Conjugate Gradient Pursuit (SWCGP) can provide a similar deblending result as the robust MP with much fewer iterations needed. For the robust MP algorithm, for the same examples, we set the maximum iterations to 500 for each window. The saving in number of iterations is significant. As we mentioned, the cost of robust Stagewise Weak Conjugate Gradient Pursuit and robust Matching Pursuit is similar for one iteration. The saving in the number of iterations contributes to the saving of time directly.

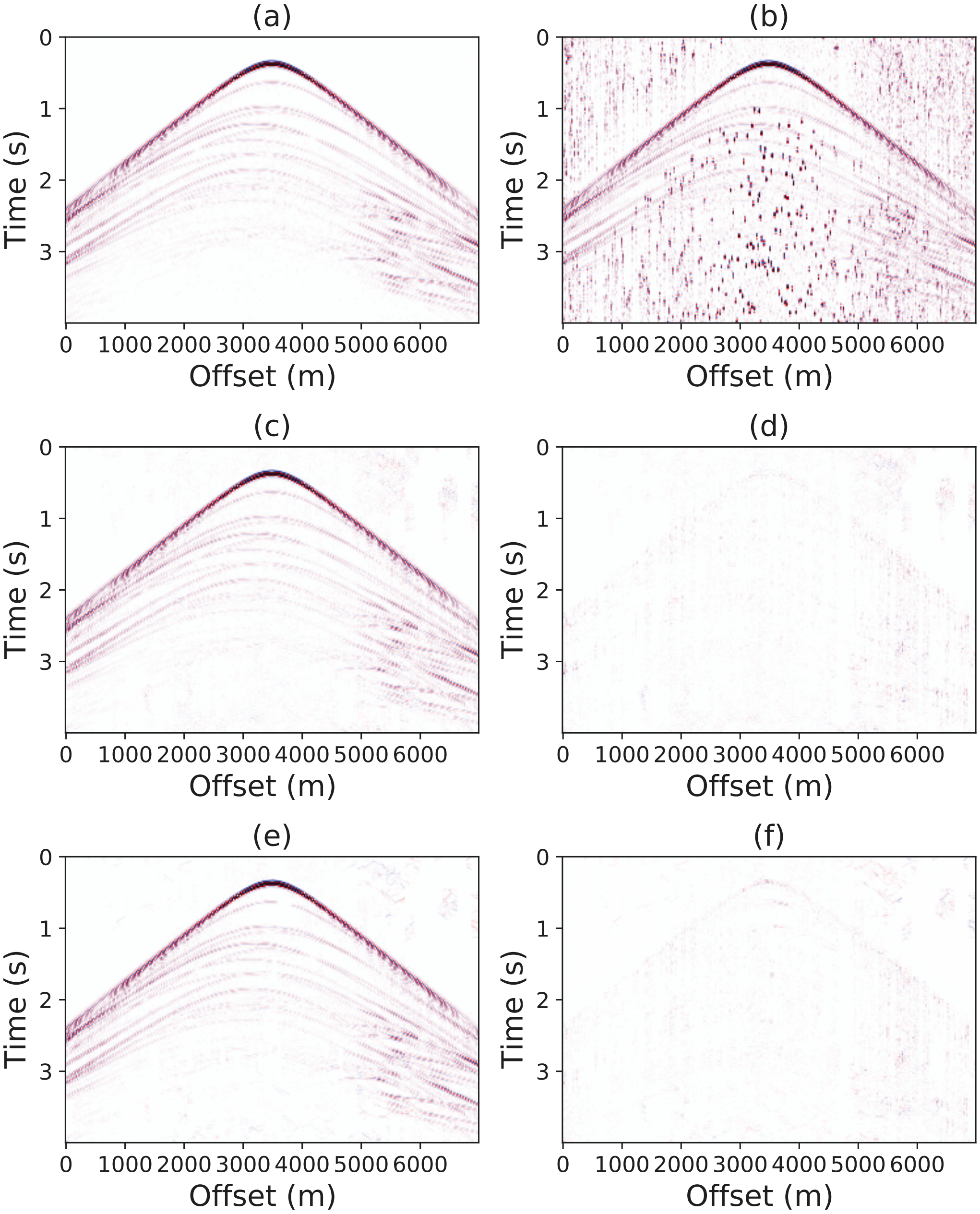

Figure 4.13: (a) One ideal common receiver gather. (b) Pseudo-deblended data common receiver gather $SNR_{in} = -1.73$ dB. (c) Deblending via robust Matching Pursuit $SNR_{out} = 17.8$ dB. (d) Ideal data minus deblended data in (c). (e) Deblending via robust SWOMP $SNR_{out} = 17.1$ dB. (f) Ideal data minus deblended data in (e).

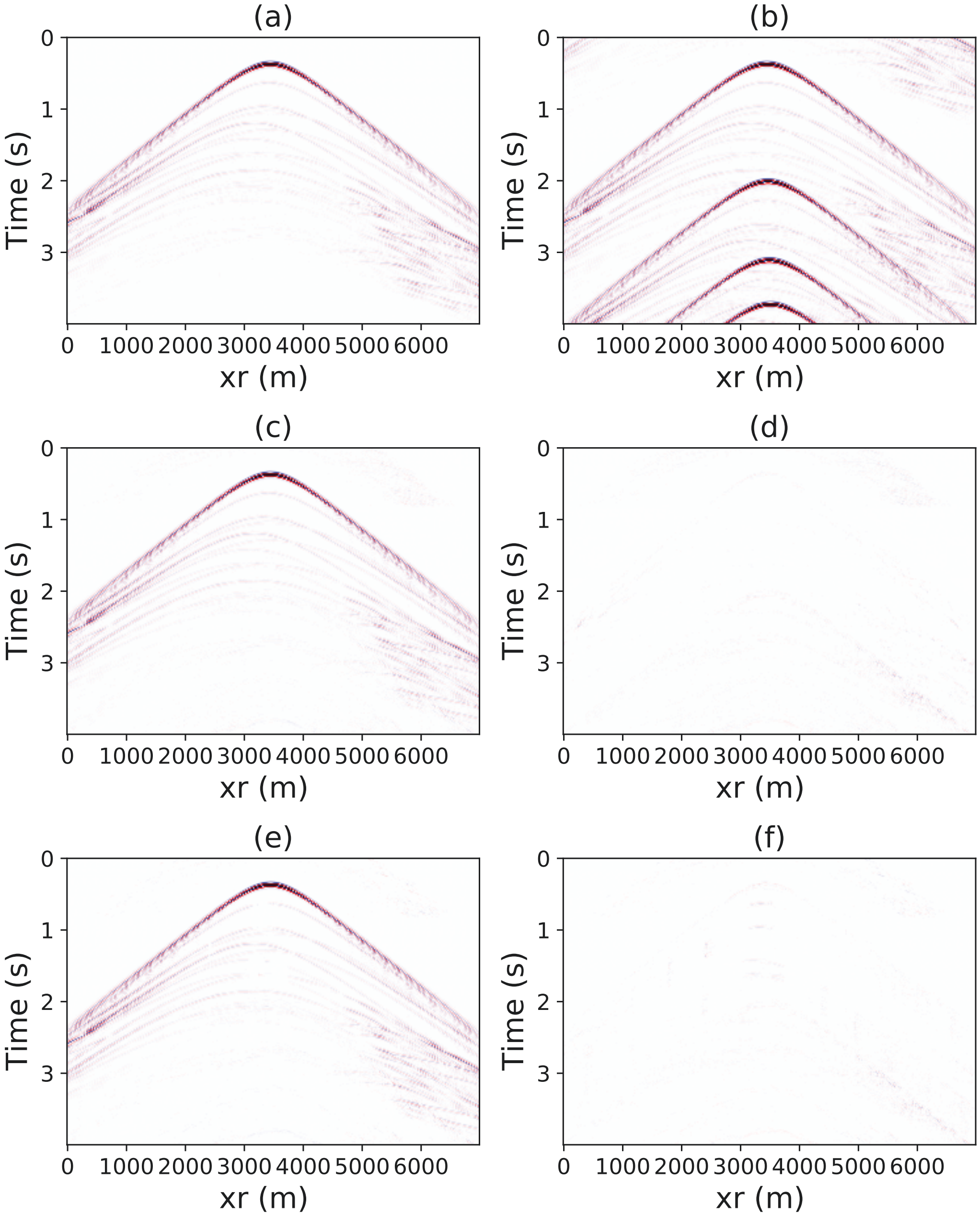

Figure 4.14: (a) One ideal common-shot section. (b) Pseudo-deblended common shot section $SNR_{in} = -3.6$ dB. (c) Deblending via robust Matching Pursuit $SNR_{out} = 18.8$ dB. (d) Ideal data minus deblended data in (c). (e) Deblending via robust SWOMP $SNR_{out} = 18.4$ dB. (f) Ideal data minus deblended data in (e).

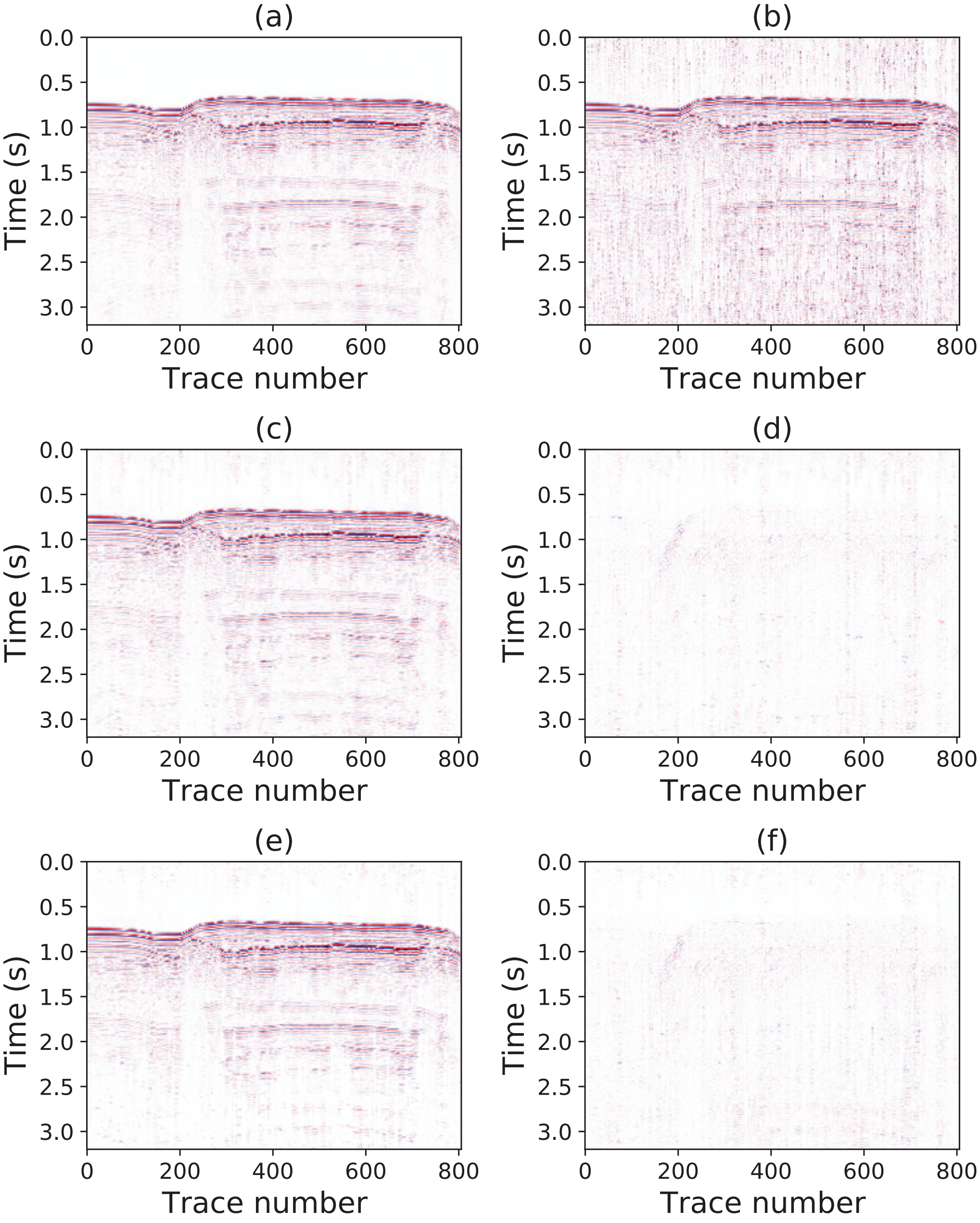

Figure 4.15: (a) One ideal common-offset gather. (b) Pseudo-deblended data common offset gather $SNR_{in} = -0.65$ dB. (c) Deblending via robust Matching Pursuit $SNR_{out} = 11.7$ dB. (d) Ideal data minus deblended data in (c). (e) Deblending via robust SWOMP $SNR_{out} = 11.3$ dB. (f) Ideal data minus deblended data in (e).

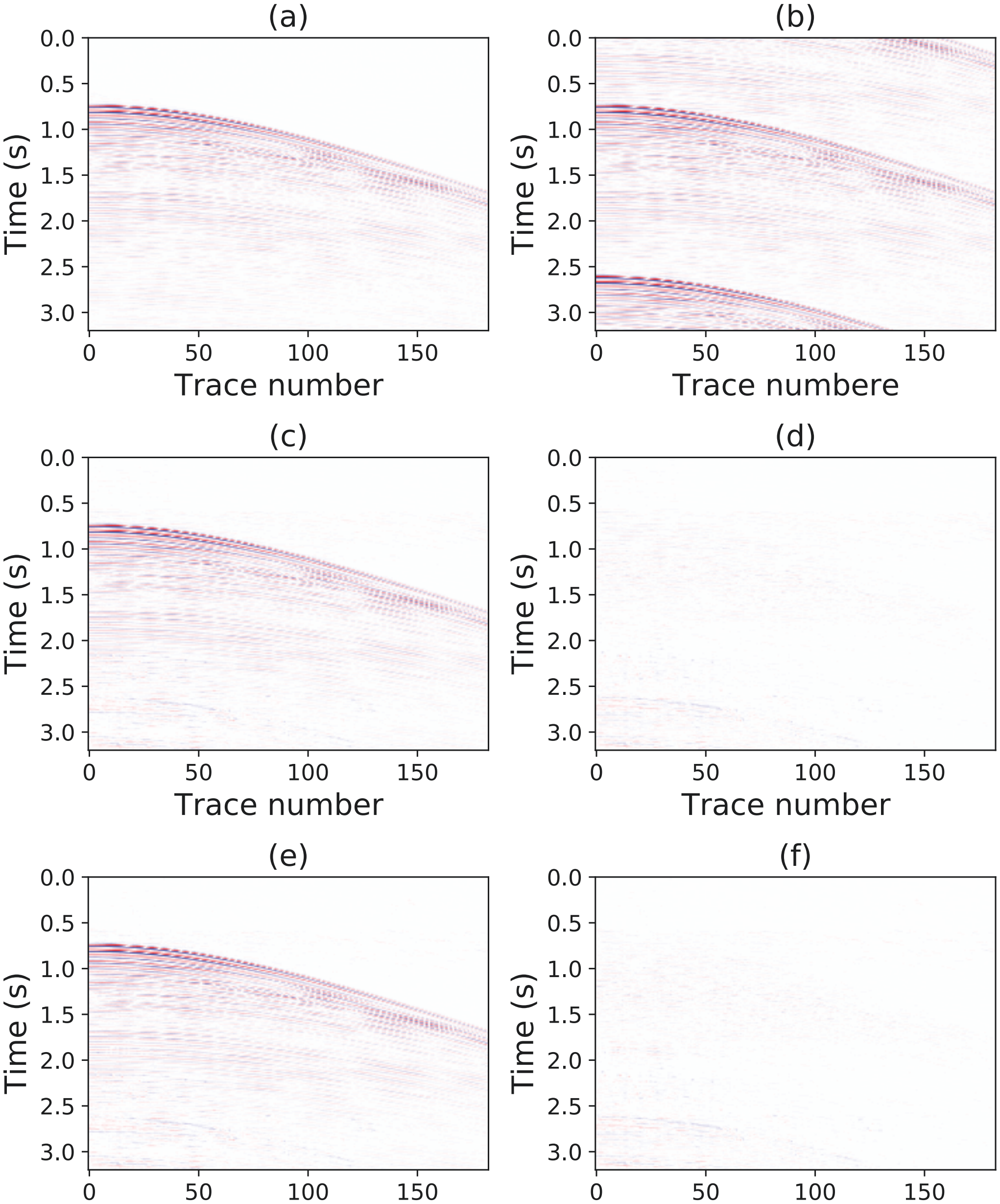

Figure 4.16: (a) One ideal common-shot section. (b) Pseudo-deblended common offset section $SNR_{in} = -1.85$ dB. (c) Deblending via robust Matching Pursuit $SNR_{out} = 13.2$ dB. (d) Ideal data minus deblended data in (c). (e) Deblending via robust SWOMP $SNR_{out} = 12.8$ dB. (f) Ideal data minus deblended data in (e).

CHAPTER 5

Conclusions

Compressive sensing and sparse reconstruction play an essential role in geophysics data acquisition and seismic signal processing. In this thesis, I focused on sparse reconstruction algorithms and applications. The generally used sparse reconstruction methods adopt an ℓ_2 norm to fit the data. These methods work quite well for noise-free data or data only contaminated with Gaussian random noise. They do not work when data are corrupted with erratic noise, such as the blending noise encountered in simultaneous source processing. Therefore, in this thesis, I become interested in making sparse approximation algorithms robust to the presence of erratic noise. I recognize that the algorithms presented in this thesis can also apply to problems of multiple suppression, seismic data interpolation, deghosting and probably many other processes that require robust sparse inversions.

In Chapter 1, I provide a brief overview of the compressive sensing theory and sparse reconstruction algorithms. These algorithms can be grouped into two major categories. One is based on convex relaxation by replacing the ℓ_0 norm regularization term with the ℓ_1 norm regularization term. The other is the greedy pursuit algorithm, which solves the problem iteratively. All these methods are only working for noise-free data or data with Gaussian random noise.

In Chapter 2, I present a review of robust sparse reconstruction strategies. I first introduce the M-estimator concept and influence functions and explained how the different M-estimators affect the cost function robustness. Then, I run a simple example with the ℓ_p norm estimator solved by the iterative reweighted least-squares (IRLS) method to show the difference between the robust and non-robust M-estimators. In the end, I also provide a review about other ways that can be used to solve the linear inverse problem with M-estimators.

The main contributions of my thesis are presented in Chapter 3 and Chapter 4. Chapter 3 proposes a newly defined ℓ_p norm inner product, which is robust to erratic noise. The latter replaces the traditional inner product in the Matching Pursuit (MP) algorithm. The MP algorithm picks the best-correlated basis atom based on the absolute value of the inner product between the residual $\mathbf{r}$ and the element of a given dictionary D. However, the traditional inner product used in the MP algorithm is defined in the ℓ_2 norm space and is very sensitive to erratic noise, and it can lead the MP to pick the wrong basis atom. Replacing the traditional inner product with our robust inner product can make the MP algorithm robust to the erratic noise. I also explained how to use the robust MP when one only has an operator instead of an explicit form matrix, which is the sparse Radon transform in our case. By adopting the robust MP method in conjunction with the Radon transform, I applied the robust MP method to the simultaneous source separation problem. Experiments show decent deblending results, but unfortunately, MP requires many iterations when applied to realistic size problems. In other words, the major limitation of the proposed robust MP algorithm for deblending is its computational cost. MP only picks and fit one coefficient in each iteration. For complex synthetics and real datasets, hundreds of thousands of coefficients need to be selected in each gather. Also, to improve results, seismic gathers are broken into many small overlapping windows. This makes the problem computational expensive because robust MP must be applied to all the windows that make all the gathers.

Therefore, in Chapter 4, I introduce robust and fast greedy pursuit algorithms. I first introduce the orthogonal matching pursuit (OMP) algorithm since all the fast greedy pursuit algorithms are based on the OMP algorithm. The primary computational costs of the OMP are coming from two parts: identification step (pick the best-correlated coefficient) and estimation step (fit the selected coefficients to the residual). Therefore, I introduce different multi-element selection strategies to reduce the total number of identification steps needed. I also propose a conjugate gradient pursuit method to minimize the computational cost of the estimation step. After many tests and comparisons, I decided to use the stagewise orthogonal matching pursuit (SWOMP) for the proposed robust, fast greedy pursuit algorithm. I use the robust SWOMP algorithm for deblending the data that was also used in Chapter 3. The main results show that, for the robust SWOMP, one can use much fewer iterations to achieve a similar deblending performance as the robust MP. For the robust SWOMP, we only use 20 iterations for each window. This value is about 500 for the robust MP used in Chapter 3.

I believe that designing fast and accurate, robust sparse reconstruction algorithms is a topic of interest for seismic processing practitioners. There are many avenues to improve these algorithms and make them applicable to large real-world scale seismic data processing problems that involve data denoising. Some problems need to be carefully examined in the future, such as proper parameter selection for robust sparse optimization problems. For instance, an objective way of accessing the number of coefficients that one must retrieve to avoid over or under-fitting when data are contaminated with erratic noise requires more research. Similarly, my thesis has adopted the Radon transform as the sparsifying transform of the problem. Other transforms could have been adopted and, in particular, one should study if adaptive transforms derived from the data themselves via, for instance, dictionary learning techniques are one way of further improving the performance of the algorithms presented in this thesis. Last, I envision also applications of the proposed methods to problems of time-frequency analysis where greedy algorithms in conjunction with localized harmonic functions can be used for robust estimation of time-evolutionary spectra with application to the study of non-stationary time series that arise in fields such as climatology, paleomagnetism and bioacoustics.

Bibliography

Abma, R., D. Howe, M. Foster, I. Ahmed, M. Tanis, Q. Zhang, A. Arogunmati, and G. Alexander, 2015, Independent simultaneous source acquisition and processing: Geophysics, **80**, WD37–WD44.

Abma, R., T. Manning, M. Tanis, J. Yu, and M. Foster, 2010, High quality separation of simultaneous sources by sparse inversion: 72nd EAGE Conference and Exhibition incorporating SPE EUROPEC 2010, European Association of Geoscientists & Engineers, cp–161.

Ahmad, F., and M. G. Amin, 2013, Through-the-wall human motion indication using sparsity-driven change detection: IEEE Transactions on Geoscience and Remote Sensing, **51**, 881–890.

Akerberg, P., G. Hampson, J. Rickett, H. Martin, and J. Cole, 2008, Simultaneous source separation by sparse radon transform, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 2801–2805.

Baraniuk, R., and P. Steeghs, 2007, Compressive radar imaging: 2007 IEEE Radar Conference, 128–133.

Baraniuk, R. G., 2007, Compressive sensing [lecture notes]: IEEE Signal Processing Magazine, **24**, 118–121.

Beasley, C. J., 2008, A new look at marine simultaneous sources: The Leading Edge,

27, 914–917.

Beck, A., and M. Teboulle, 2009, A fast iterative shrinkage-thresholding algorithm for linear inverse problems: SIAM Journal on Imaging Sciences, **2**, 183–202.

Berkhout, A., and D. Verschuur, 2006, Focal transformation, an imaging concept for signal restoration and noise removal: Geophysics, **71**, A55–A59.

Berkhout, A. J., 2008, Changing the mindset in seismic data acquisition: The Leading Edge, **27**, 924–938.

Blumensath, T., and M. E. Davies, 2008a, Gradient pursuits: IEEE Transactions on Signal Processing, **56**, 2370–2382.

———, 2008b, Iterative thresholding for sparse approximations: Journal of Fourier Analysis and Applications, **14**, 629–654.

———, 2009, Stagewise weak gradient pursuits: IEEE Transactions on Signal Processing, **57**, 4333–4346.

Bobin, J., J. Starck, and R. Ottensamer, 2008, Compressed sensing in astronomy: IEEE Journal of Selected Topics in Signal Processing, **2**, 718–726.

Boyd, S., N. Parikh, E. Chu, B. Peleato, and J. Eckstein, 2011, Distributed optimization and statistical learning via the alternating direction method of multipliers: Foundations and Trends® in Machine Learning, **3**, 1–122.

Candes, E. J., J. Romberg, and T. Tao, 2006, Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information: IEEE Transactions on Information Theory, **52**, 489–509.

Candes, E. J., and M. B. Wakin, 2008, An introduction to compressive sampling: IEEE Signal Processing Magazine, **25**, 21–30.

Candès, E. J., 2008, The restricted isometry property and its implications for compressed sensing: Comptes Rendus Mathematique, **346**, 589 – 592.

Chen, S. S., D. L. Donoho, and M. A. Saunders, 1998, Atomic decomposition by basis pursuit: SIAM Journal on Scientific Computing, **20**, 33–61.

Chen, Y., C. Caramanis, and S. Mannor, 2013, Robust sparse regression under adversarial corruption: Proceedings of the 30^{th} International Conference on Machine Learning, PMLR, 774–782.

Chen, Y., S. Fomel, and J. Hu, 2014, Iterative deblending of simultaneous-source seismic data using seislet-domain shaping regularization: GEOPHYSICS, **79, no. 5**, V179–V189.

Cheng, J., and M. D. Sacchi, 2015, Separation and reconstruction of simultaneous source data via iterative rank reduction: Geophysics, **80**, V57–V66.

Combettes, P. L., and V. R. Wajs, 2005, Signal recovery by proximal forward-backward splitting: Multiscale Modeling & Simulation, **4**, 1168–1200.

Dai, W., and O. Milenkovic, 2009, Subspace pursuit for compressive sensing signal reconstruction: IEEE Transactions on Information Theory, **55**, 2230–2249.

Daubechies, I., M. Defrise, and C. De Mol, 2004, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint: Communications on Pure and Applied Mathematics, **57**, 1413–1457.

Deans, S. R., 2007, The radon transform and some of its applications: Courier Corporation.

Donoho, D. L., 1995, De-noising by soft-thresholding: IEEE Transactions on Information Theory, **41**, 613–627.

———, 2006, Compressed sensing: IEEE Transactions on Information Theory, **52**, 1289–1306.

Donoho, D. L., Y. Tsaig, I. Drori, and J. Starck, 2012, Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit: IEEE Transactions on Information Theory, **58**, 1094–1121.

Eldar, Y. C., 2008, Generalized sure for exponential families: Applications to regularization: IEEE Transactions on Signal Processing, **57**, 471–481.

Figueiredo, M. A. T., R. D. Nowak, and S. J. Wright, 2007, Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems: IEEE Journal of Selected Topics in Signal Processing, **1**, 586–597.

Galatsanos, N. P., and A. K. Katsaggelos, 1992, Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation: IEEE Transactions on image processing, **1**, 322–336.

Gan, S., S. Wang, Y. Chen, and X. Chen, 2016, Simultaneous-source separation using iterative seislet-frame thresholding: IEEE Geoscience and Remote Sensing Letters, **13**, 197–201.

Golub, G. H., M. Heath, and G. Wahba, 1979, Generalized cross-validation as a method for choosing a good ridge parameter: Technometrics, **21**, 215–223.

Haacke, R., G. Hampson, and B. Golebiowski, 2015, Simultaneous shooting for sparse obn 4d surveys and deblending using modified radon operators: 77th EAGE Conference and Exhibition 2015, European Association of Geoscientists & Engineers, cp–451.

Hampel, F. R., E. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, 1986, Robust statistics: the approach based on influence functions. (ID: unige:23238).

Hampson, D., 1986a, Inverse velocity stacking for multiple elimination, *in* SEG Technical Program Expanded Abstracts 1986: Society of Exploration Geophysicists, 422–424.

———, 1986b, Inverse velocity stacking for multiple elimination: Canadian Journal of Exploration Geophysics, **22**, 44–55.

Hennenfent, G., and F. J. Herrmann, 2008, Simply denoise: Wavefield reconstruction via jittered undersampling: GEOPHYSICS, **73**, V19–V28.

Herrmann, F. J., 2010, Randomized sampling and sparsity: Getting more information from fewer samples: GEOPHYSICS, **75**, WB173–WB187.

Holland, P. W., and R. E. Welsch, 1977, Robust regression using iteratively reweighted least-squares: Communications in Statistics-theory and Methods, **6**, 813–827.

Huo, S., Y. Luo, and P. G. Kelamis, 2012, Simultaneous sources separation via multidirectional vector-median filtering: Geophysics, **77**, V123–V131.

Ibrahim, A., and M. D. Sacchi, 2014, Simultaneous source separation using a robust radon transform: Geophysics, **79**, V1–V11.

Jeong, W., C. Tsingas, and M. S. Almubarak, 2020, A numerical study on deblending of land simultaneous shooting acquisition data via rank-reduction filtering and signal enhancement applications: Geophysical Prospecting.

Ji, J., 2006, Cgg method for robust inversion and its application to velocity-stack inversion: Geophysics, **71**, R59–R67.

Kabir, M. N., and D. Verschuur, 1995, Restoration of missing offsets by parabolic radon transform 1: Geophysical Prospecting, **43**, 347–368.

Kassam, S. A., and H. V. Poor, 1985, Robust techniques for signal processing: A survey: Proceedings of the IEEE, **73**, 433–481.

Kim, S., K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, 2007, An interior-point method for large-scale ℓ_1-regularized least squares: IEEE Journal of Selected Topics in Signal Processing, **1**, 606–617.

Kontakis, A., and D. Verschuur, 2017, Using a hybrid focal: curvelet transform for deblending, *in* SEG Technical Program Expanded Abstracts 2017: Society of Exploration Geophysicists, 4903–4908.

Kuchment, P., 2013, The radon transform and medical imaging: SIAM.

Laska, J. N., M. A. Davenport, and R. G. Baraniuk, 2009, Exact signal recovery from sparsely corrupted measurements through the pursuit of justice: 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers, 1556–1560.

Li, X., 2012, Compressed sensing and matrix completion with constant proportion of corruptions.

Liu, B., and M. D. Sacchi, 2004, Minimum weighted norm interpolation of seismic records: Geophysics, **69**, 1560–1568.

Lustig, M., D. Donoho, and J. M. Pauly, 2007, Sparse mri: The application of compressed sensing for rapid mr imaging: Magnetic Resonance in Medicine, **58**, 1182–1195.

Malioutov, D. M., M. Cetin, and A. S. Willsky, 2005, Homotopy continuation for sparse signal representation: Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005., v/733–v/736 Vol. 5.

Mallat, S., 2008, A wavelet tour of signal processing, third edition: The sparse way, 3rd ed.: Academic Press, Inc.

Mallat, S. G., and Z. Zhang, 1993, Matching pursuits with time-frequency dictionaries: IEEE Transactions on Signal Processing, **41**, 3397–3415.

Moore, I., B. Dragoset, T. Ommundsen, D. Wilson, C. Ward, and D. Eke, 2008, Simultaneous source separation using dithered sources, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 2806–2810.

Moore, I., R. Fletcher, C. Beasley, and C. Castellanos, 2016, Data studies of simultaneous source separation using robust linear algebra: SEG Technical Program Expanded Abstracts 2016, 4623–4627.

Needell, D., and J. Tropp, 2009, Cosamp: Iterative signal recovery from incomplete and inaccurate samples: Applied and Computational Harmonic Analysis, **26**, 301 – 321.

Needell, D., and R. Vershynin, 2008, Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit: Foundations of Computational Mathematics, **9**, 317–334.

———, 2010, Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit: IEEE Journal of Selected Topics in Signal Processing, **4**, 310–316.

Nguyen, N. H., and T. D. Tran, 2011, Robust lasso with missing and grossly corrupted observations.

———, 2013, Exact recoverability from dense corrupted observations via ℓ_1-minimization: IEEE Transactions on Information Theory, **59**, 2017–2035.

Nyquist, H., 1928, Certain topics in telegraph transmission theory: Transactions of the American Institute of Electrical Engineers, **47**, 617–644.

Osborne, M., B. Presnell, and B. Turlach, 2000, A new approach to variable selection in least squares problems: IMA Journal of Numerical Analysis, **20**, 389–403.

Pati, Y. C., R. Rezaiifar, and P. S. Krishnaprasad, 1993, Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, 40–44 vol.1.

Pham, D., and S. Venkatesh, 2012, Improved image recovery from compressed data contaminated with impulsive noise: IEEE Transactions on Image Processing, **21**, 397–405.

———, 2013, Efficient algorithms for robust recovery of images from compressed data: IEEE Transactions on Image Processing, **22**, 4724–4737.

Razavi, S. A., E. Ollila, and V. Koivunen, 2012, Robust greedy algorithms for compressed sensing: 2012 Proceedings of the 20^{th} European Signal Processing Conference (EUSIPCO), 969–973.

Russell, B., D. Hampson, and J. Chun, 1990a, Noise elimination and the radon transform, part 1: The Leading Edge, **9**, 18–23.

———, 1990b, Noise elimination and the radon transform, part 2: The Leading Edge, **9**, 31–37.

Sacchi, M. D., and T. J. Ulrych, 1995, High-resolution velocity gathers and offset space reconstruction: Geophysics, **60**, 1169–1177.

Sacchi, M. D., T. J. Ulrych, and C. J. Walker, 1998, Interpolation and extrapolation using a high-resolution discrete fourier transform: IEEE Transactions on Signal Processing, **46**, 31–38.

Scales, J. A., and A. Gersztenkorn, 1988, Robust methods in inverse theory: Inverse Problems, **4**, 1071–1091.

Schonewille, M., Z. Yan, M. Bayly, and R. Bisley, 2013, *in* Matching pursuit Fourier interpolation using priors derived from a second data set: 3651–3655.

Shannon, C. E., 1949, Communication in the presence of noise: Proceedings of the IRE, **37**, 10–21.

Studer, C., and R. G. Baraniuk, 2013, Stable restoration and separation of approximately sparse signals.

Studer, C., P. Kuppinger, G. Pope, and H. Bölcskei, 2011, Recovery of sparsely corrupted signals.

Tibshirani, R., 1996, Regression shrinkage and selection via the lasso: Journal of the Royal Statistical Society. Series B (Methodological), **58**, 267–288.

Trad, D., 2009, Five-dimensional interpolation: Recovering from acquisition constraints: Geophysics, **74**, V123–V132.

Trad, D., T. Ulrych, and M. Sacchi, 2003a, Latest views of the sparse radon transform: Geophysics, **68**, 386–399.

———, 2003b, Latest views of the sparse Radon transform: Geophysics, **68**, 386–399.

Trad, D. O., T. J. Ulrych, and M. D. Sacchi, 2002, Accurate interpolation with high-resolution time-variant radon transforms: Geophysics, **67**, 644–656.

Tropp, J. A., 2004, Greed is good: algorithmic results for sparse approximation: IEEE Transactions on Information Theory, **50**, 2231–2242.

Tropp, J. A., and A. C. Gilbert, 2007, Signal recovery from random measurements via orthogonal matching pursuit: IEEE Transactions on Information Theory, **53**, 4655–4666.

Wang, H., G. Li, and G. Jiang, 2007, Robust regression shrinkage and consistent variable selection through the lad-lasso: Journal of Business & Economic Statistics, **25**, 347–355.

Wen, F., P. Liu, Y. Liu, R. C. Qiu, and W. Yu, 2016, Robust sparse recovery for compressive sensing in impulsive noise using $\ell_p - norm$ model fitting: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 4643–4647.

Wright, J., and Y. Ma, 2009, Dense error correction via l1-minimization: 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 3033–3036.

Xu, S., Y. Zhang, D. Pham, and G. Lambaré, 2005, Antileakage fourier transform for seismic data regularization: Geophysics, **70**, V87–V95.

Yang, J., and Y. Zhang, 2011, Alternating direction algorithms for ℓ_1-problems in compressive sensing: SIAM Journal on Scientific Computing, **33**, 250–278.

Zeng, W., H. C. So, and X. Jiang, 2016, Outlier-robust greedy pursuit algorithms in ℓ_p-space for sparse approximation: IEEE Transactions on Signal Processing, **64**, 60–75.

Zoubir, A. M., V. Koivunen, Y. Chakhchoukh, and M. Muma, 2012, Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts: IEEE Signal Processing Magazine, **29**, 61–80.

Zwartjes, P., and M. Sacchi, 2007, Fourier reconstruction of nonuniformly sampled, aliased seismic data: Geophysics, **72**, V21–V32.

APPENDIX A

Derivation of equation 3.10

For a vector $\mathbf{x}$ of length N its p norm $\|\mathbf{x}\|_p$ is defined as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{N} |x_i|^p\right)^{\frac{1}{p}} \,. \tag{A.1}$$

The computation of the l_p inner product (equation (13)) requires minimizing the cost

$$J = \|\mathbf{r} - a\,\mathbf{g}\,\|_p^p \,. \tag{A.2}$$

Taking the derivative of J with respect to the unknown a and equating the derivative to zero leads to

$$\begin{aligned} \frac{dJ}{da} &= \frac{d}{da}\sum_i |r_i - ag_i|^p \\ &= p\sum_i |r_i - ag_i|^{p-1}\,\mathrm{sgn}(r_i - ag_i)g_i^* \\ &= 0\,, \end{aligned} \tag{A.3}$$

where sgn is the signum function which for a complex number z is given by

$$\mathrm{sgn}(z) = \frac{z}{|z|}\,. \tag{A.4}$$

Then equation A.3 becomes

$$\frac{dJ}{da} = \sum_i |r_i - ag_i|^{p-2}(r_i - ag_i)g_i^* = 0, \tag{A.5}$$

which leads to

$$\sum_i r_i w_i g_i^* = a \sum_i g_i\, w_i\, g_i^*, \tag{A.6}$$

or

$$a = \frac{\sum_i r_i w_i g_i^*}{\sum_i g_i\, w_i\, g_i^*}, \tag{A.7}$$

were the weights are approximated by the following expression

$$w_i = \frac{1}{|r_i - ag_i|^{2-p} + \epsilon}\,. \tag{A.8}$$

It is clear that A.7 can be written in the following form

$$a = \frac{\mathbf{g}^H \mathbf{W} \mathbf{r}}{\mathbf{g}^H \mathbf{W} \mathbf{g}}. \tag{A.9}$$

A few iterations (typically 4 to 5) are sufficient to estimate the parameter a with initial value $a_o = 0$.